COMPUTING FOR HOME AND BUSINESS

INTERFACE AGE™

OCTOBER 1980

$2.50/CANADA/MEXICO $3.00

Micros in education: making the climb

The electronic checkbook is here

02651

10 0 71486 02651 DIR

Build your own telephone dialer

Hard and Fast...

...Bulk Storage from Industrial Micro Systems

THE NEW MODEL 16

The new Industrial Micro Systems Model 16 Hard Disk Subsystem is a "fixed-removable" high speed, bulk storage device providing from 32 megabytes (32 million characters) to 96 megabytes of on-line storage for the Industrial Micro Systems 8000 or Series 5000 microcomputer systems. The Model 16 includes a credenza enclosure that provides a quiet, strong and attractive package for office or industrial applications where large memory is required. The Model 16 also includes a fully buffered DMA S-100 bus controller for fast and easy interfacing.

WINCHESTER TECHNOLOGY WITH BUILT-IN BACKUP

The Model 16 includes a 16 megabyte removable cartridge and a 16, 48, or 80 megabyte fixed media that employs Winchester 3340 technology. Files and programs may be copied between the fixed media and the removable cartridge for fast, easy backup and archival storage.

FAST ACCESS

The interface between the Model 16 hard disk and the Industrial Micro Systems computer is provided by the Hard Disk Controller. The Hard Disk Controller utilizes Direct Memory Access (DMA) for fast data transfer with minimum processor intervention. The maximum data transfer rate is 1.2 megabytes per second and the controller fully buffers the data, a sector at a time, to and from the disk. **Available in 220 V, 50 HZ Versions**

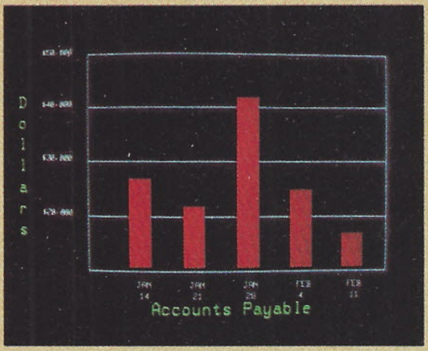
Now you don't have to look hard for fast computing power. Contact your Industrial Micro Systems Dealer today.

INDUSTRIAL MICRO SYSTEMS

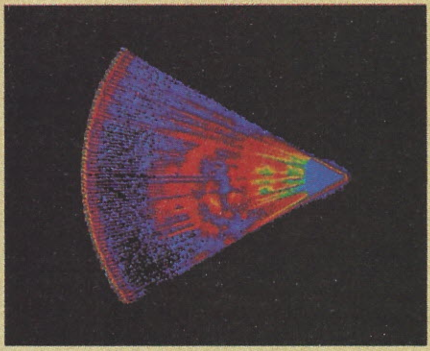
Marketing
628 N. Eckhoff, Orange, CA 92668
(714) 978-6966

Manufacturing
2800 Lockheed Way, Carson City, NV 89701
(702) 883-7611

CIRCLE INQUIRY NO. 23



Management Information Display


Ultrasonic heart sector scan

High-resolution display with alphanumerics

Get the professional color display that has BASIC/FORTRAN simplicity

LOW-PRICED, TOO

Here's a color display that has everything: professional-level resolution, enormous color range, easy software, NTSC conformance, and low price.

Basically, this new Cromemco Model SDI* is a two-board interface that plugs into any Cromemco computer.

The SDI then maps computer display memory content onto a convenient color monitor to give high-quality, high-resolution displays (756 H x 482 V pixels).

When we say the SDI results in a high-quality professional display, we mean **you can't get higher resolution than this system offers in an** NTSC-conforming display.

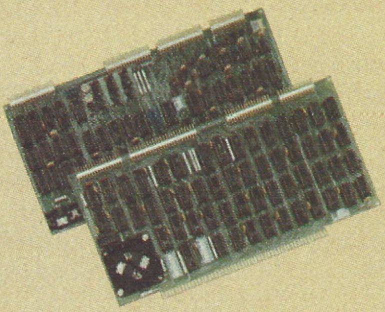
The resolution surpasses that of a color TV picture.

BASIC/FORTRAN programming

Besides its high resolution and low price, the new SDI lets you control with optional Cromemco software packages that use simple BASIC- and FORTRAN-like commands.

Pick any of 16 colors (from a 4096-color palette) with instructions like DEFCLR (c, R, G, B). Or obtain a circle of specified size, location, and color with XCIRC (x, y, r, c).

*U.S. Pat. No. 4121283


Model SDI High-Resolution Color Graphics Interface

HIGH RESOLUTION

The SDI's high resolution gives a professional-quality display that strictly meets NTSC requirements. You get 756 pixels on every visible line of the NTSC standard display of 482 image lines. Vertical line spacing is 1 pixel.

To achieve the high-quality display, a separate output signal is produced for each of the three component colors (red, green, blue). This yields a sharper image than is possible using an NTSC-composite video signal and color TV set. Full image quality is readily realized with our high-quality RGB Monitor or any conventional red/green/blue monitor common in TV work.

Model SDI plugs into Z-2H 11-megabyte hard disk computer or any Cromemco computer

DISPLAY MEMORY

Along with the SDI we also offer an optional fast and novel **two-port** memory that gives independent high-speed access to the computer memory. The two-port memory stores one full display, permitting fast computer operation even during display.

CONTACT YOUR REP NOW

The Model SDI has been used in scientific work, engineering, business, TV, color graphics, and other areas. It's a good example of how Cromemco keeps computers in the field up to date, since it turns any Cromemco computer into an up-to-date color display computer.

The SDI has still more features that you should be informed about. So contact your Cromemco representative now and see all that the SDI will do for you.

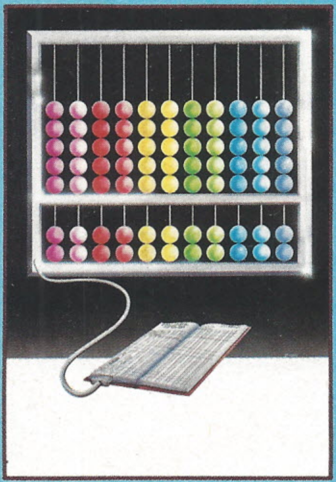
Cromemco
incorporated
280 BERNARDO AVE., MOUNTAIN VIEW, CA 94040 • (415) 964-7400
Tomorrow's computers today
CIRCLE INQUIRY NO. 7

VOL. 5, ISSUE 10
SINCE DECEMBER 1975

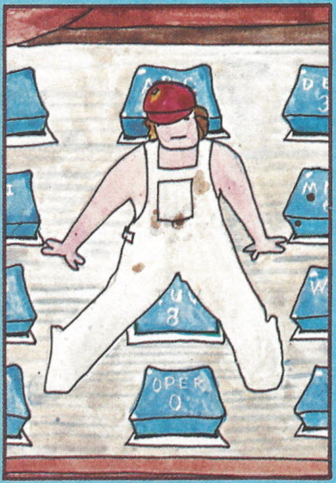
OCTOBER 1980

INTERFACE AGE™

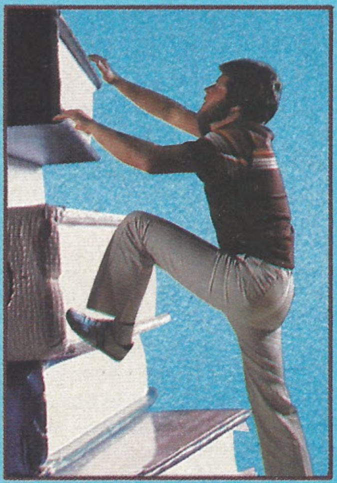
COMPUTING FOR HOME AND BUSINESS APPLICATIONS



Binary Cents 70



Automatic Telephone Dialer . 74



Cover photo by . . . . . Don May
Model . . . . Jim Ramsey

FEATURES

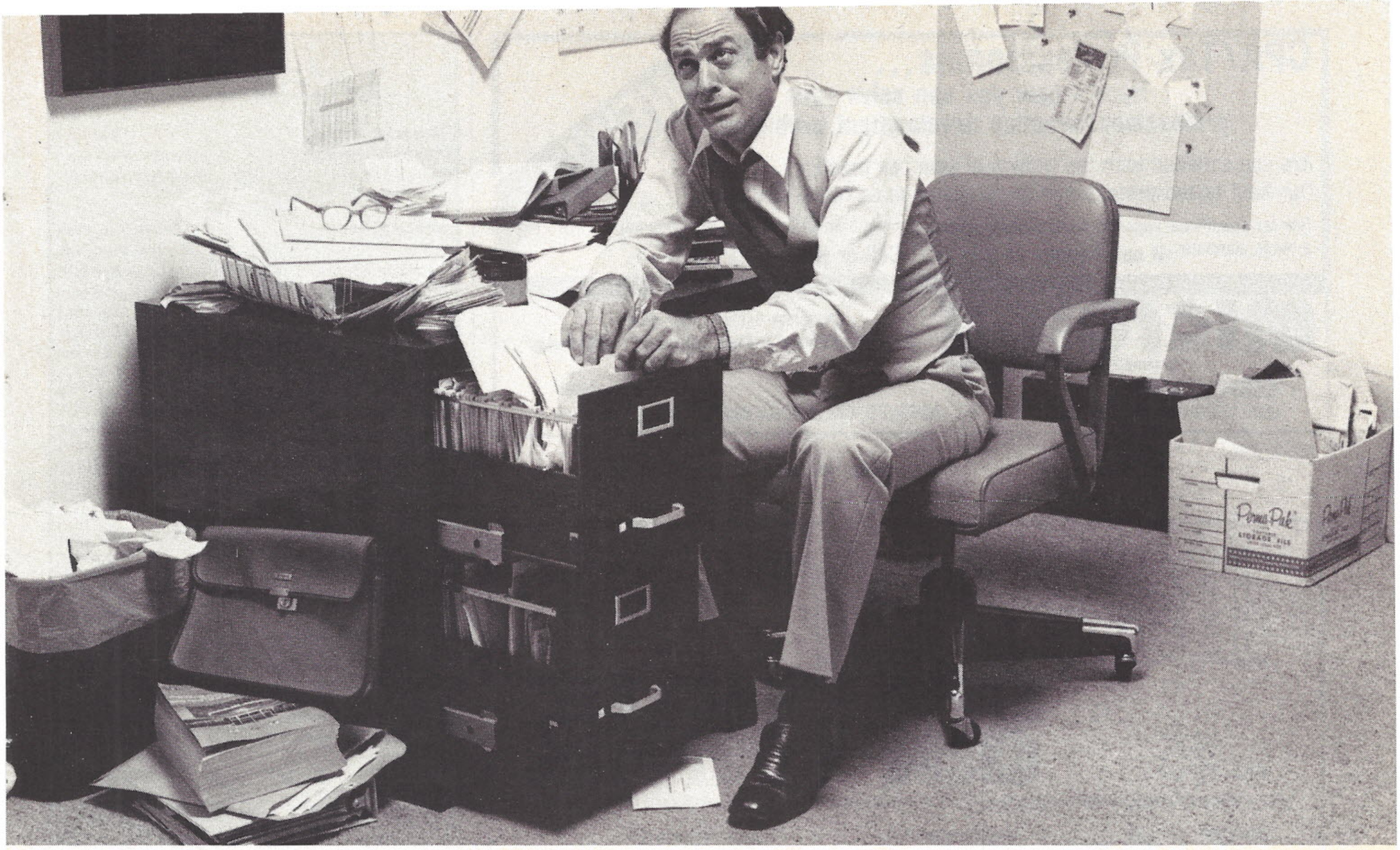
COLUMNS

DEPARTMENTS

Contact authors of monthly columns by writing to them at INTERFACE AGE, P.O. Box 1234, Cerritos, CA 90701 in care of their respective columns.

INTERFACE AGE Magazine, published monthly by McPheters, Wolfe & Jones, 16704 Marquardt Ave., Cerritos, CA 90701. Subscription rates: U.S. $18.00, Canada/Mexico $20.00, all other countries $28.00. Make checks payable in U.S. funds drawn on a U.S. bank. Opinions expressed in by-lined articles do not necessarily reflect the opinion of this magazine or the publisher. Mention of products by trade name in editorial material or advertisements contained herein in no way constitutes endorsement of the product or products by this magazine or the publisher. Circulation Department, (213) 926-9540.

INTERFACE AGE Magazine is catalogued in the Library of Congress, Classification No. QA75.5.155. USPS No. 528150. ISSN Publication No. 0147-2992. Membership in Audit Bureau of Circulations applied for.
POSTMASTER: Please send change of address form 3579 and undelivered copies to **INTERFACE AGE** Magazine, 16704 Marquardt Ave., Cerritos, CA 90701. Controlled circulation postage paid at Olive Branch, Mississippi and Artesia, California.



Record keeping problems? Our CCA Data Management System solves them easily.

Having information at your fingertips can make your job a whole lot easier. And that's what the CCA Data Management System is all about.

With this Personal Software™ package and an Apple II™ or TRS-80™ disk system, it will be far easier to keep inventories, customer lists, accounts receivable and payable records, patient histories and many more items.

In fact, you can use the CCA DMS for all of your data management needs, rather than buying (expensive) or writing (time consuming) separate programs for each application. That's because DMS lets you create your own filing systems, adapting itself to the types of records you keep. You specify the number and names of each data field—without any programming.

With DMS keeping all of your records, you only have to learn how to use one system. That's easier, too. It's menu driven, with plenty of prompts to help you create files and add, update, scan, inspect, delete, sort, condense and print data. Our comprehensive 130-page step-by-step instruction manual even provides complete "how to" inventory and mailing list applications so you can start processing immediately.

DMS is a very powerful system, with more file and record storage capacity than other data base programs on the market.

*Apple is a trademark of Apple Computer, Inc.; TRS-80 is a trademark of the Radio Shack Div. of Tandy Corp.

And it also gives you greater data handling flexibility. To customize DMS, write add-on BASIC programs that read or write DMS files and perform any kind of processing you want.

You can sort and print your data in nearly any form of report and mailing label you want. Sort data by up to 10 fields for zip code, balance due, geographic location or whatever. And print reports with subtotals and totals automatically calculated.

Apple DMS has two additional features. Its ISAM search method helps you find any item on a diskette within 10 seconds. And it's Data Interchange Format Program allows you to move DMS files into our Apple VisiCalc™ program—the "electronic worksheet"—for powerful, flexible calculating.

Ask your Personal Software dealer to show you how easy computerized record keeping is. To locate the nearest dealer, contact Personal Software Inc., 408/745-7841, 1330 Bordeaux Drive, Sunnyvale, CA 94086.

CIRCLE INQUIRY NO. 48

PERSONAL SOFTWARE

INTERFACE AGE

Robert S. Jones Publisher
Nancy A. Jones Executive Publisher
Mike Antich Publication Director

Editorial

Managing Editor **Frank Vaughan**
Editor **Michael Panchak**
Assistant Editors **Kathy Tekawa, Les Spindle**
Contributing Editors **Alfred Adler**
Bob Albrecht
Al Baker
R. W. Bemer
Roger Edelson
Tom Fox
Roger C. Garrett
Carl Heintz
Elliott MacLennan
Alan R. Miller
Jim Schreier

Production

Production Manager **Terri Ledesma**
Art Director **Fino Ortiz**
Artist **Samantha Lee**
Typographers **Melody A. Martens**
Suzi Pippin

Administration

Accounting Supervisor **Kay Soto**
Accounting Assistants
Mary Ann Lower, Sharon Spolar
Assistant-to-the-Publisher **Eva Lewis**
Circulation Manager **Colin Cato**
Circulation Assistants
Vicky Goodman, Randy Morris
Publication Assistants
Cheryl Johnston, Lilly Lisa, Doris Riopel

Advertising

New England Region **Dick Green**
7 Lincoln St., Wakefield, MA 01880
(617) 245-9105

Eastern Region **John Sensenstein**
20 Community Pl., Ste. 140, Morristown, NJ 07960
(201) 267-3032

Southeast Region **Harry Dill**
3938 Sussex Avenue, Charlotte, NC 28210
(704) 552-1004

Midwest Region **Al Gravenhorst, Steve Skinner**
5901 N. Cicero Ave., Chicago, IL 60646
(312) 545-8621

Texas Region **Mitch Mohanna**
2312 Canyon Valley Trail, Plano (Dallas), TX 75023
(214) 596-1139

Western Region **Mike Antich**
P.O. Box 1234, Cerritos, CA 90701
(213) 926-9544

Japan **Tomoyuki Inatsuki**
Trade Media Japan Inc., R. 212 Azabu Hts., 1-5-10, Roppongi, Minato-ku, Tokyo 106
Telephone: (03) 585-0581 Telex: J28208

Interface Age Europe

Director, European Operations **H.L. Grohmann**
Dahlienstr. 4, D-8011 Munchen-Vaterstetten
West Germany
Telephone: 08106/7396

International Newsstand Distribution

Director **Lew Ullian**
Orberstrasse 38, D-6000 Frankfurt/M. 61
West Germany
Telephone: (0611) 44 77 90/41 84 80

International/Domestic Retail Circulation

(213) 926-9544 **Mary Ann Lower**

MEMBER OF THE WESTERN PUBLICATIONS ASSOCIATION wpa

AMERICAN SOCIETY OF BUSINESS PRESS EDITORS MEMBER

16704 Marquardt, Cerritos, CA 90701
(213) 926-9544
TWX (910) 583-1412

EDITOR'S NOTEBOOK

Radio Shack announces 1981 product line

Rumors are interesting and, like a bottle of fine wine, people tend to hold and savor particular types. Such has been the case with Radio Shack.

Since the introduction of the TRS-80 model II, Radio Shack has been the subject of enormous attention by the rumor mongers. We've all heard that the model I was going out of production, that TRSDOS was being discontinued in favor of CP/M, that a $200 color machine was in production and filling warehouses somewhere in Texas, or that a TRS-90 was in the wings.

Well put your favorite rumor to bed. Radio Shack has made what is probably the largest computer-related product announcement in its history. The company has proved one point beyond a doubt—that the "top-level" information the rumor mongers have been spreading isn't as top-level as it's touted.

The meat of Radio Shack's announcement consisted of:

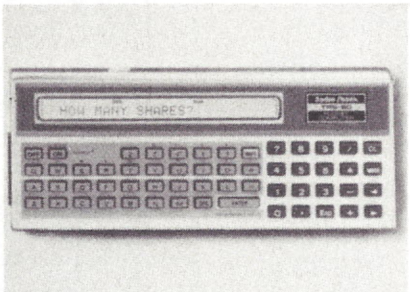
- TRS-80 pocket computer
- TRS-80 model III
- TRS-80 color computer
- Videotex terminal
- Printer/plotter
- Daisy wheel printer II
- Line printer VI

Before getting into the products, I think it is important to put something into perspective. Radio Shack is rarely given credit for being a real part of the computer industry. There has been a tendency to consider the company as a merchandiser that happens to sell a low-cost, mass-market microcomputer. I would suspect that stereo industry people consider Radio Shack a merchandiser which happens to sell low-cost, mass-market stereo systems. This line of reasoning could apply to the television, two-way radio or any other market the company is in.

The fact that Radio Shack has sold more than 200,000 small computers has not propelled it into the ranks of a "real" computer company.

When the full impact of its 1981 computer line sinks in, anyone who looks upon this Fort Worth company as "only" a merchandiser will be making a big mistake. Radio Shack appears to be in the computer industry to stay, and the remainder of the industry can no longer bury its head in the sand.

Probably the biggest surprise was its introduction of the pocket computer. At



the summer CES in Chicago, handheld computers by Nixdorf and Panasonic/Quasar caused quite a stir. To find Radio Shack willing to venture into this new segment of the industry is startling and, I believe, underscores the company's commitment. The pocket computer is similar in appearance to the language translators popular several years ago. It has a 24-character liquid crystal display with English language prompting and Basic programming. The Basic is similar, but not identical, to level I Basic.

The battery-powered unit includes 1.9K-bytes of RAM, weighs six ounces and is less than seven inches long. The unit comes with a cassette interface that will sell for $49 more than the handheld's $249.95 price tag. Presently no other peripherals are available, but Radio Shack has committed itself to manufacturing a printer.

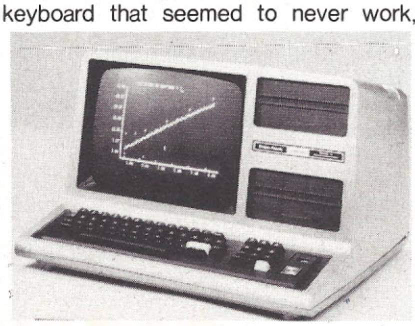
This unit is not of Radio Shack manufacture and is not made in the United States. Sharp Electronics seems to be the manufacturer of the unit, which is physically identical (except the nameplate) with the Sharp PC1211 handheld, and the Sharp PC1210 handheld, which has slightly less memory. The cassette interface seems to be the CE121 Sharp unit. Sharp reportedly will not begin marketing in the U.S. until mid-1981, which gives Radio Shack an edge.

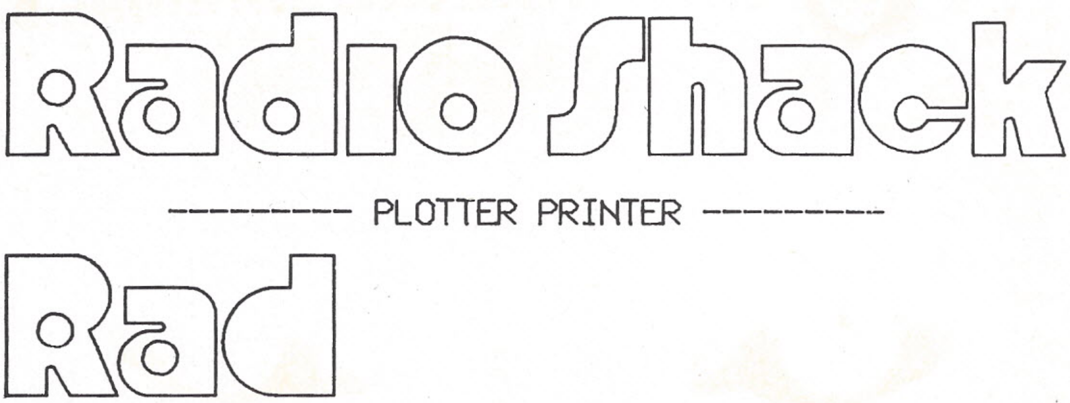
The lack of peripherals is somewhat a surprise. The Nixdorf and Panasonic/Quasar units have fairly complete lines, including acoustic couplers, expansion units, cash register tape-type printers and power supplies. While Radio Shack is willing to commit at least a printer, I was unable to obtain a confirmation as to what type the company is looking to.

Model III

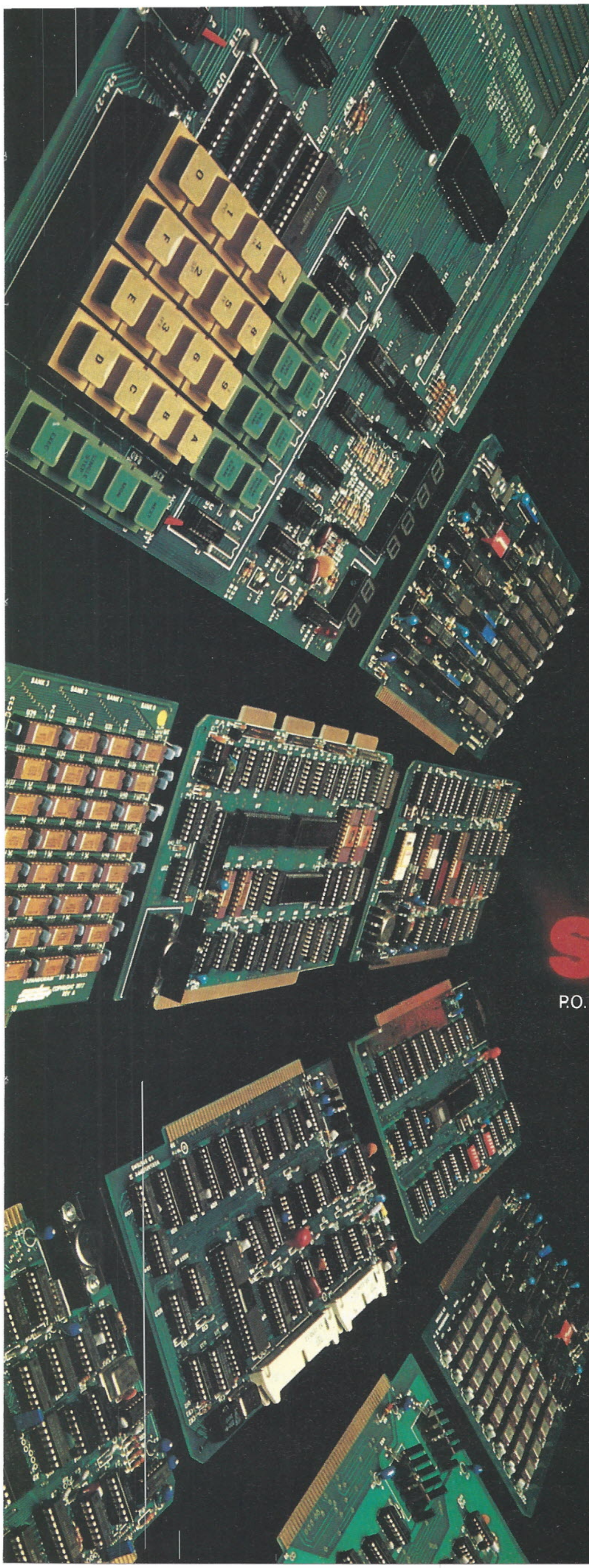
The stellar announcement, at least in Radio Shack's eyes, was the model III. This is the machine that firmly plants the company in the computer industry—and will serve as its credentials in the future.

Who can forget the model I, with the keyboard that seemed to never work,





Sample plotter output



enough wires and cables to thrill an electrician, the strange-looking expansion interface, and the accompanying giggles from the rest of the computer industry. That scenario has not been repeated. The model III has gotten rid of that cheap, plastic look, shows some excellent styling, and doesn't have a visible cable—except to peripherals.

In fact, the unit resembles the Apple III in many respects. Certainly, both feature themselves as integrated units with keyboard, memory, and diskette storage capabilities in one.

This Radio Shack offering has a handsome sculpted keyboard, a variety of memory sizes ranging from 4K to 48K, and up to two onboard diskettes with a configuration limit of four.

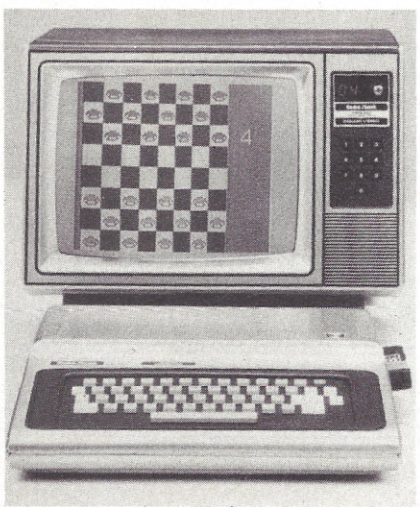
The model III is available in three versions. The level I system with 4K of RAM prices at $699. It will reportedly run with all model I, level I software programs as it uses the level I Basic.

A second version with 16K of RAM and model III Basic will sell for $999 and features upper and lower case.

The third version, dubbed the desktop business computer, has 32K of RAM, two double-density 40-track disk drives, an RS-232 serial interface and sells for $2,495. It can be expanded to the full 48K and four disk drives. One of its high points is the ability to read existing model I disks, which immediately makes all of that software out there available for immediate use. The model III Basic is "compatible with most TRS-80 model I programs," according to the company. Interestingly enough, the company still denies that production on the model I has been stopped.

Colors, at last

The long-awaited color unit seemed dwarfed in this announcement by the previous two products, but if color is what you want, then color is what you get. For "under $400," it is offering a 4K RAM, 8K ROM unit with a 53-key typewriter-type keyboard, 1,500 baud cassette interface and RS-232 serial interface. The color computer provides a screen format of 16 lines, 32 characters-per-line and graphics arrays from 32 by 64 to 196 by 256. The machine uses plug-in color program pack, and the initial offering includes: personal finance, math, bingo, chess, checkers, music, football, pinball and quasar commander.



The program packs sell for $29.95 to $39.95. Joystick controls with push-button triggering are $24.95. A 16K RAM unit is available as an expansion option for $218 or as a complete unit, upping the system price tag to $599. A matching 13-inch color TV receiver is available for $399.

Looking very much like the color computer, but bearing a different keyboard, is the Videotex terminal that will sell for $399 and be available in November. It includes a direct-connect telephone modem and 4K of memory, and can be used for accessing networks such as Micronet.

Radio Shack is quick to point out that the Videotex terminal is just that...a terminal, and despite its resemblance to the color computer, is not a computer.

The daisy wheel printer is just that, a daisy wheel printer, with some unique application of plastics and a price tag of $1,960—well below the average.

The printer plotter, at $1,460, is bound to become quite popular. It features multidirectional printing and has an average print speed of CPS on 7.5-inch lines.

The Lineprinter VI, a wide bed 100 CPS printer featuring bidirectional 9 by 7-dot print head, can use between 4-inch and 15-inch paper, has a built in character self-test and limited graphics capability. Made overseas, the unit carries a $1,160 price tag, is being made exclusively for Radio Shack and uses a Centronics-type interface.

Incidentally, one of the chief criticisms against Radio Shack has been the way it refuses to acknowledge anyone else in the industry. Its marketing policies in company-owned stores highlight this view, since it doesn't even allow non-Radio Shack publications to be sold. That wall may be about to crack.

Along with the Videotex announcement, Radio Shack also advised us that it will be selling a videotex software package for Apple that will retail for $29.95, and are seriously considering an offering of the package for other popular microcomputers. **FV**

"We're back together again."

For those of you who come to us for our quality brand of computer peripheral products and components, CMC Marketing has come up with three good reasons to get us together again. For good.

We've developed three complete small business computer systems in addition to our dependable line of components that brought us together in the first place. At prices that should keep us together for a long, long time.

A basic CMC system, like the economical Series 100, featuring the powerful Z-80A microprocessor and the standard S-100 bus is available for less than $9,350.00. And this system utilizes the industry standard CP/M® operating system compatible with most CP/M® software. All CMC systems are expandable to include multi-terminal capability, on-line communications and extended hard disk options, should your applications needs change or grow.

If you'd like more information about the new CMC small computer systems, call us toll free at: 1-800-231-9833. After all, it's time we got back together.

®CP/M is a registered trademark of Digital Research Corp.

CMC Marketing
COMPUTER SYSTEMS

CMC Marketing Corporation/10611 Harwin Suite 406/Houston, Texas 77036

LETTERS

Court case legalizes stealing?

I was not surprised at the findings reported on in your July Editor's Notebook.

Having been to Apple and TRS-80 computer clubs where the main attraction was duplicating the latest version of Star Trek, and hearing that the Microsoft Basic compiler would soon be available from so-and-so at a reasonable five finger discount, nothing much surprises me, nor should most adults be surprised at finding out that immorality exists in the world.

When dealing with humans, two basic axioms should be observed:

> Paying nothing always costs less than paying something.
>
> Innocence is equal to not being caught.

In general I consider ethics codes a response to the fact that immorality exists; but codes without enforcement have little real meaning. Laws must be enforced to be laws. It is easy to forget that wars are won by attrition, not by good intentions and never by talk.

In trying to enforce any law, you have to first catch someone. And since a lot of "stealing" is done at computer club meetings and by groups of friends and groups of employees, the cost of finding the villain can be more than the product is worth.

If I understand the recent Data Cash vs. J.S.&A. ruling, the courts have held that computer programming on magnetic media does not have the copyright protection given to printed material. So we are getting closer to a victimless society. You can't steal a program in the public domain because you already own it.

Ronald C. Wagener
Norfolk, VA

Hearing things

There are whispers around that someone is about to hit the market with a cassette recorder for under $250 that has 8-track parallel input. I assume that it is much like a tape reader, only much, much faster. Best of all it will offer approximately eight times the storage capacity per tape as a standard cassette recorder. It seems to me that the electronics for parallel I/O is a lot simpler than serial I/O, and therefore cheaper to build. But who makes a cheap 8-track head? Do any of your readers know who is about to put this recorder on the market? If so, would they share this info with me?

S.B. Wahlberg
P.O. Box 502
Silverado, CA 92676

Scheduling software needed

Who has TRS-80 software for PERT/CPM scheduling? I'm told by experts that it's possible to use a microcomputer for this purpose. It's time that someone came up with a low-cost computer package for use in project management. When I find one, I'll give it exposure to the persons who come to my project management seminars. Make the contact for me, and I'll put the show on the road.

S. F. Love
Advanced Professional
Development Institute
5519 Carpenter Ave.
No. Hollywood, CA 91607

Review provokes strong rebuke

I was quite upset to find a favorable review of *Software Interpreters for Microcomputers* by Thomas C. McIntire (IA May 1980). This is one of the worst books of its kind that I have read. The author is *not* a good programmer. He espouses using all kinds of little tricks to "save" small amounts of memory and to "increase" execution speed at the expense of producing incomprehensible code. He seems completely unaware that execution speed is far more sensitive to the overall approach to the problem than whether subroutines take a little more or a little less time. For example, in chapter 14 in the discussion of execution speed, he ignores the fact that hash tables can give enormous increases in execution speed. He espouses packing two 4-bit counters in a byte of memory, eight 1-bit flags in a byte, and using one memory location for different things at different times, all in an attempt to save memory space. Not only does this trickery lead to code that is extremely difficult to understand and modify, but in addition it creates the possibility that obscure bugs may cause occasional system crashes, and that these bugs will be virtually impossible to find, as anyone who has worked on programs like this will testify. . .And McIntire acknowledges that what he does is considered to be bad programming practice, but he says, "It is considered to be fundamentally correct here. . .that anything that works correctly is legitimate discipline. . ." The programming practices espoused by McIntire have been tried and proven to be bad. It does not matter how fast a program is and how little space it occupies in the memory if it is not correct. I think that any programmer will agree that there are enough unreliable programs. We don't need books encouraging people to write more.

Another annoyance is that the book is set in a sans serif font that is impossible to read quickly or easily: the reader's eyes get lost on the page. But perhaps this was a good book in which to make such an exception.

Richard L. Miller
Cambridge, MA

Some added notes on Zilog

I am grateful for the mention given my PDS by Alan Miller in his August article. However, a few points regarding the unannounced Z-80 instructions should be made: First, we should thank Mr. D.R. Lunsford for making these 94 additional operations known to us. Second, I agree with Dr. Miller that software for distribution should avoid the undocumented instructions. The individual user may favor these instructions if his processor has the capability. I have not found a Z-80 processor—Zilog or second source—which does not handle the extended instructions. Third, while it is regrettable that ZSID cannot handle these instructions, that fact cannot preclude their use when it may be advantageous. The 'debug' component of PDS fully recognizes the enhanced instruction set.

Allen Ashley
Pasadena, CA

Documentation criticized

As I recently received a Z-89, the article by Tom Fox in August was of great interest. I fully agree that the hallmark of the Z-89 is the terminal, and was in general agreement until he praised the documentation.

In the weeks that I have worked with the system I have found endless frustrations with the software documentation and sizable holes in the hardware manual. Section 0 of the HDOS manual is excellent for the beginner whether business man or hobbyist. After learning to use one-copy, the reader is forced into section 1 which is written as a reference for experienced programmers. I found sections 1 through 5 useless until the local Heath store gave me the name of a systems analyst who was gracious enough to spend a Saturday helping me.

If Zenith is serious about marketing the Z-89 to businessmen, it must cookbook the manual more, supply much more interesting information and improve its service.

Robert C. Thompson
Houston, TX

Is Expandability the Reason Over 200,000 Smart Buyers Chose TRS-80? Or is It Price? Or . . .

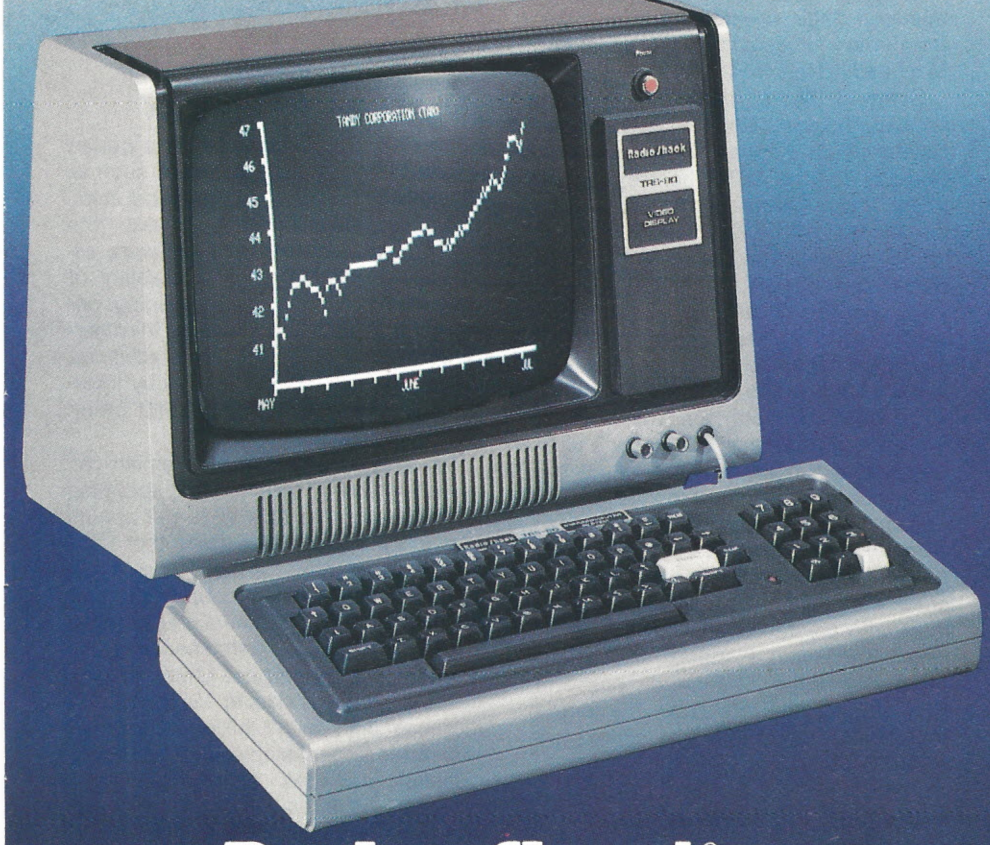
Maybe these customers came to Radio Shack for their computer because they knew they could get nationwide service? Or buy a Model I micro starter system for $499? Or because Radio Shack started the boom in reliable little computers? But come they did. And coming they still are!

"Who Wants One That Can't Grow?"

This question was asked by Radio Shack's president before TRS-80™ went to market. So we made memory expandable from 4K to 48K RAM (in 16K leaps). We made ROM expandable and BASIC upgradable. We provided for the addition of disk drives, printers, acoustic couplers and other peripherals. We have a great new upper/lower case conversion, and a SCRIPSIT™ package that makes TRS-80 Model I a great bargain in Word Processing.

"Be Better Than Competition!"

"Not merely cheaper," he said, "not just more popular," he said, "Better!" That includes software, hardware, service, monitor size, keyboards, salesmen, store fronts, peripherals and RAMs. If you own a TRS-80, let us show you how to better your system and its benefits. If you're still on the sidelines . . . shouldn't you play with us? After all, Radio Shack and TRS-80 have FAST DELIVERY FROM STOCK or a very short wait. Most (if not all) the competitors are much slower players!



Expansion Interface

The hub of an expanded TRS-80 system. Holds up to 32K more memory! Has interfaces for printer, 4 disk drives, second cassette. Also takes RS-232C interface card. **$299**

Save $100

Now Only

Reg. 499.00 **$399** Each

Mini-Disk Drives

Quality engineered disk drives let you store and retrieve data and programs fast. Sale ends 9/30/80. Order today! **$399**

Line Printer II

Quality for less! Prints 100 cps on 80 columns. Accepts 9⅞" fanfold, 8½" roll paper or single sheets. **$799**

For more reasons, send for our free computer catalog!

NAME ______

STREET ______

CITY ______ STATE ______ ZIP ______

PHONE NO. ______

Radio Shack

Dept. CMA-81-A-22
1300 One Tandy Center
Ft. Worth, TX 76102

Radio Shack®

The biggest name in little computers™

CIRCLE INQUIRY NO. 49

A DIVISION OF TANDY CORPORATION, FORT WORTH, TEXAS 76102

Retail prices may vary at individual stores and dealers.
Some Computer equipment may require special order.

Study forecasts steady micro growth despite economic slowdown

The microprocessor market will more than triple by 1983 according to a research study by International Data Corp. Shipments will swell from last year's $738 million to $2.3 billion by year-end 1983. The primary microprocessor market—8-bit and 16-bit loose microprocessor chips (including support chips)—was worth some $598 million in 1979. IDC projects a growth rate of 36% per year, reaching $2.05 billion in 1983. This estimate allows for the economic slowdown during 1980. In fact, IDC reports that the recession is already impacting the micro market with declining delivery times. In 1981, when the economic picture brightens, a sharp upturn is predicted in microprocessor usage.

The importance of support chips is apparent when one examines the shipment breakdown. Micro chips had shipments of $119 million in 1979, while support and memory chips had shipments of $479 million—or roughly 65% of the total market last year. By 1983, these chips will account for 77% of the total market. Note, however, that the growth predicted is due mainly to increased usage of memory chips. I/O chips are declining in market share as many of their former functions are now included on processor chips.

In addition to the market for loose chips, microprocessor chips and associated memories may also be incorporated into boards—either single board computers or dedicated function boards. The added value of these boards (excluding the value of the components on the boards) was $140 million in 1979, and is expected to grow at a 23% annual rate to $316 million in 1983.

Firm simply refuses to let the defense rest

After it won the case brought against it last August by Michael Shrayer Software, Vector Graphic of Westlake Village, CA, has turned around and filed a cross complaint against Shrayer alleging libel and interference with business relationships, and a second suit against Shrayer and several others for trade libel.

The judgement was granted to Vector after the court found that Shrayer had failed to prove its allegation that Vector had wrongfully copied the Electric Pencil II word processing program.

Performance evaluations prepared on top business micro systems

Within the past few years, the business microcomputer has reached maturity. Not only is there a wide variety of computer systems available for the small business, but with available application program packages, they can be employed almost immediately.

Unlike the larger minicomputers, micros are marketed through retail stores—a relatively new concept in computer merchandising. MIC, a data processing publication company, has issued evaluations of the major business systems: Radio Shack TRS-80 I and II, IBM 5110 and 5120, Vector Graphic Systems, Alpha Micro, Ohio Scientific, Rexon, North Star, Cromemco, Apple II and Burroughs B90.

Each of these evaluations is an in-depth analysis of the computer product, its hardware, and software.

This set of business evaluations is available for $75 ($85 outside the U.S. and Canada) from Management Information Corp., 140 Barclay Center, Cherry Hill, NJ 08034, (609) 428-1020.

The history of programming put on audio and video tapes

Cassettes highlighting the early technical developments of 13 programming languages, and the factors that influenced their creation, are available from the Association of Computing Machinery, New York City.

Some of the key personalities in the field are included: Alan Perlis and Peter Naur on Algol, Kenneth Iverson on APL, Douglas Ross on APT, Thomas Kurtz on Basic, Jean Sammet on Cobol, John Backus on Fortran, Geoffrey Gordon on GPSS, Charles Baker on JOSS, Jules Schwartz on Jovial, John McCarthy on Lisp, George Radin on PL/I, Kristin Nygaard and Ole-Johan Dahl on Simula, and Ralph Griswold on Snobol.

In addition to the cassette programs, all excerpted from the 1978 ACM History of Programming Language Conference, a 2-hour program of all the technical sessions, a program of the opening session (introduced by Jean Sammet and keynoted by Grace Hopper), a 2-hour program of humorous anecdotes from the conference banquet with Bernard Galler as master of ceremonies, and a special slide-show, consisting of the 2-hour audio-cassette program of excerpts from the entire technical program and 174 slides, are also available.

Keyboards still dominate as communication means

Keyboards will remain the major means for people to communicate to machines for at least the next 5 years, and the increasing flood of keyboard-equipped games, terminals and word processors will buoy the market for keyboards. According to International Resource Development, shipments of keyboards will exceed $100 million in 1980, and this figure will at least double by 1985. Although the report attaches "great significance" to the rapid development of voice recognition equipment, the IRD report expects the keyboard market to be largely unaffected by this technology during the current decade.

Membrane technology, in which the keyboard is created from several layers of specially-configured conducting and non-conducting sheets, is the fastest-growing keyboard technology. Membrane keyboards are popular in applications where dirt or fluids are encountered, such as in process control and household appliances, and they can be mass-produced inexpensively. Although much of the future market for membrane keyboards will be in entirely new applications, such as keyboard-controlled ignition and door locks for automobiles, membranes are expected to take some market share away from other established keyboard technologies, such as capacitive, inductive and Hall effect.

Europe: a changing software and service market

Western Europe is entering a period of rapid change. Market forces such as the small business system and minicomputer, new DDP equipment, the wider range of packaged software offerings, and the greater availability of processing services is increasing demand for processing services, third party software, and professional advisors, according to a research study by International Data Corp. A near-$10 billion market is forecast by 1982.

Processing services will remain the fastest growing sector, user spending currently growing at a 16% per annum clip (only slightly less than the 17% growth forecasted for the U.S. market). Spending on remote processing services will swell by some 28%, totaling $3.6 billion in 1982. Batch services, accounting for the largest chunk of processing service dollars in 1978, will only increase 4%, to $2 billion. Currently, IBM is the largest single supplier of processing services in Europe with revenues of $394 million in 1978. (In the U.S., because of antitrust restrictions, until 1979 IBM could not and does not now even participate in this market.) Most of IBM's closest competitors are French-based companies.

"For reliable data storage, I recommend systems with Shugart disk drives."

Tom Knight, President—
Nycom, Los Altos, California

"The last thing you need when you put your personal computer or small business system to work is a disk drive that you can't rely on. If the drive quits, your system is out of business."

That's why more and more manufacturers and dealers depend on Shugart disk drives for reliable data storage. These professionals don't want disk drive problems any more than you do. Shugart has a large family of drives, too—in all sizes and capacities to suit your system storage needs. For the smaller system, the original 5¼-inch Minifloppy™ stores 250 to 500 kilobytes (single or double-sided)—that's about 50 to 100 pages of printed material. Our single and double-sided 8-inch floppys store 800 to 1600 kilobytes. And for systems that need a larger data base, our 8-inch or 14-inch fixed disk drives store from 5 to 58 megabytes. No other manufacturer offers such a wide variety of disk storage for personal computer and small business systems.

Word processing, general business, accounting—big system or small, you can rely on Shugart drives. We're known as the Headstrong company for good reason. We're Headstrong about reliability, quality, and value. Ask your dealer. He knows us.

Rely on the Headstrong Company.

Shugart

475 Oakmead Parkway, Sunnyvale, California 94086

TM—Minifloppy is a trademark of Shugart Associates.

Computers in the Playground

SESAME PLACE

by Kathy Tekawa

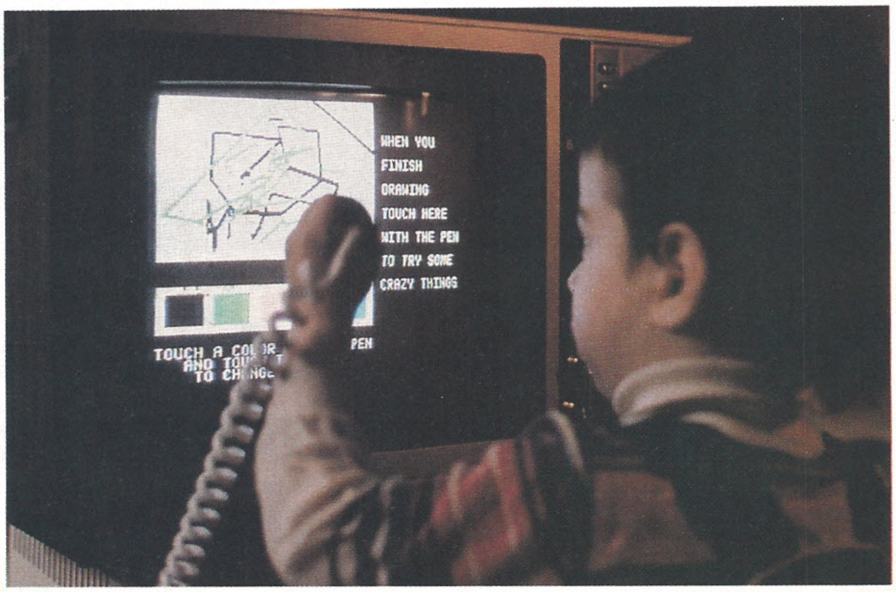
An educational playground, with playmates such as Big Bird, Oscar the Grouch and the Cookie Monster, is using microcomputers to explode the myth that learning is dull, boring and painstaking for children. The spot is Sesame Place located in Langhorne, PA and it has borrowed heavily from the children's TV show, Sesame Street, to bolster the "learning and fun" concept in a completely interactive environment.

Located within Sesame Place, a creation of the Children's Television Workshop and Busch Entertainment Corp., is a 5,700-sq.-ft. Computer Gallery which houses a collection of computer games targeted for youngsters 3-13. .although nearly anyone would enjoy playing with these lively systems.

Those who enter this electronic playground are greeted by a colorful array of 68 terminals mounted on iron pipes reminiscent of giant Tinker Toys. Sesame Place opened last July. Since then parents have flocked in to see for the first time how effective micros can be for their youngsters. Besides having fun, their children learned about space and gravity, motion, sound and conversation, illusions, art and patterns.

Before the opening of Sesame Place, computer games were tested extensively, using children, who have no problem expressing their true feelings about the project's effectiveness.

Most of the testing was conducted by Steven Gass, assistant director of research. "A lot of research was matching the game to the need of the child according to his age group. For instance, a preschooler loves exploratory games, while the teenager or adult prefers problem-solving or logic games. We had to consider each age group's needs and program them into the software," he explains. "Most of the software was designed by us, but some were modifications of existing programs."



An electronic pointer is used to draw multicolored designs.

A major concern for those in charge of the Computer Gallery was how to give the audience instant access without making them uncomfortable. "We had to ask ourselves, how are we going to overcome the standards of traditional software? How are we going to use simple instructions. . .

without heavy technical terms? Computer buffs assume that, if there's a blinking cursor, you input then return. This isn't routine information to the general public. Every time you want someone to input a 'return' you have to tell him."

According to Joyce Hakansson, computer games coordinator, "it's exciting to see adults learn with their children. Most of us never had any interaction with computers when we grew up, so we are novices as much as our kids. It's the child who often takes that first step to show a parent how the system works.

"Adults are usually much more fearful of making a mistake, especially in family experiences. They prefer that the child try everything first. A funny thing happens," she says. "The parent prompts the child to try using the computer as he watches over the child's shoulder, then he starts making suggestions. He'll say, why don't you try this or try that first—soon what's really happening is the adult is playing the game and the poor child is just doing what the parent says."

Computers made to order

All the micros in the gallery are extensively modified 48K Apples. The brick red terminals with green front plates and keyboards maintain the "playful" look found throughout the park.

The custom designed keyboards differ from conventional keyboards in that they are absolutely flat. . .and $^{50}/_{1,000}$-inch thick. Sesame Place designed its own graphic overlays to go on top.

According to Steven Gensler, president of Unicorn Engineering,

Custom designed terminals include a simplified keyboard that differs from a conventional model with a flat surface and a touch-sensitive switching panel. Large keys make it easy for even the youngest visitor to use.

Music, reading, writing, math and science are among many subjects youngsters learn about through colorful games and puzzles. The assortment is designed to appeal to every youngster, from the novice to the electronic game buff.

the firm which designed the keyboards, they are simple to use. "The keyboard was originally designed for the handicapped, then turned out to be ideal for children because of its large keys (1-inch square) and simple design possibilities."

The alphabet on the keyboard runs sequentially from A-Z. "We aren't teaching typing, just allowing children to interact with a computer. Besides, most children don't know how to type. We felt they might be fearful if they could not find the letters," Hakansson says. Also, there is a pallet of colors on the keyboard for the user to push if the computer asks what color he wants. Rather than typing in 'red,' he simply pushes the red key. There's also a 'yes' and a 'no' key, a space, erase and the numbers.

"An advantage to this type of keyboard is that it is software defined, so it lends itself to graphics," Hakansson explains. "For instance, one of our programs is a dial-a-muppet. It has a touch-tone telephone and the child picks the muppet he wants to talk to. There are different pictures of muppets on the telephone and the user chooses by pressing a picture. A voice comes on while a drawing of the muppet appears on the screen."

The main objective to this whole computer project, according to Hakasson, was to integrate learning

Continued on Page 120

A NEW MASTERPIECE IN PRINTERS

The MODEL 800 MST is certainly pleasing to look at, but its true beauty lies beneath the surface. A glimpse at its features reveals why it is rapidly becoming the most sought after printer in the world . . .

- Four standard interfaces:
  - RS-232 (15 baud rates)
  - Centronics compatible parallel
  - IEEE-488
  - 20ma current loop
- Six line densities: 64, 72, 80, 96, 120, 132
- 100 CPS at all six densities
- Unidirectional or bidirectional printing
- Sixteen horizontal and ten vertical tabs
- Elongated characters in all six densities
- 1920 character buffer
- Uses either perforated or roll paper
- Fully adjustable tractors to 9½″
- Auto self-test
- Up to 10 character fonts
  - Standard 96 character ASCII
  - User defined character font
  - Provision for up to eight additional fonts
- Dot resolution graphics in six densities
- Variable line spacing control from 0 to 64 dots in half-dot increments
- Auto form-feed for any form length at any line spacing
- Heavy-duty all aluminum chassis
- 110vac or 220vac, 50/60Hz.
- 100 million character printhead
- Measures only 15″ wide, 3″ high, and 11″ deep
- Weighs only 15 lbs.

. . . . . but maybe its most attractive feature is the price . . . . . $699.00.

base 2 inc. P.O. BOX 3548 FULL., CAL. 92634 / (714) 992-4344

CIRCLE INQUIRY NO. 2

Jurisprudent Computerist

By Elliott MacLennan
Attorney at Law

Tax Deductibility of Computer Software

A software purchaser is sometimes a creature to be pitied. Where he purchases software only, the federal investment tax credit is inapplicable because the IRS maintains that software is intangible property, and only tangible personal property qualifies. Contrariwise, many states have classified software, and not just its media, as tangible property triggering the imposition of sales tax, use tax, value-added taxes, personal property tax or some other species concocted in the brain of a creative tax administrator.

Although an investment credit is unavailable, a deduction for software in the year purchased is, so stated the court in a recent federal appellate case originating from an Idaho tax court case.

Here the court held, and the federal appeals court agreed, that software is deductible in the year purchased as opposed to being written off over its useful life, the latter position being espoused by the IRS. The two positions, useful life amortization vs. current deductibility clashed head-on with the taxpayer winning a clear victory.

In the Idaho case, the IRS argued that the purchase of computer programs and accompanying operating manuals provided only future economic benefits of an unknown duration and, therefore, must be capitalized and ratably deducted or amortized over their useful life. The IRS also argued that by purchasing the software, the taxpayer corporation created a new and separate asset which must be capitalized, not currently deducted.

The tax court and federal appellate court decided that the purchasers of the software did not create a new and separately identifiable asset but merely acquired a means to operate its business in a more cost efficient way. In addition to cost efficiency, the two courts felt that software expenses would be a recurring charge to taxpayer's business.

I believe that the decided opinions of the two courts were correct because, by way of example, the purchase by a business of an application program designed to centralize and coordinate inventory is not the creation of a new and separate asset. Rather, it is the replacement and upgrading of a human inventory control system. Will software charges for a computerized control system be recurring? That depends on a host of several factors unique to a given business application. Is it not reasonable to assume that modification of software is mandated when the manner of physical inventory storage and shipment is altered due to market conditions, warehouse space availability and the like? Perhaps it would be simpler to generalize and say that the better acquainted a business gets with its computerization, the more customized it will seek to make its software.

Software is routine expense

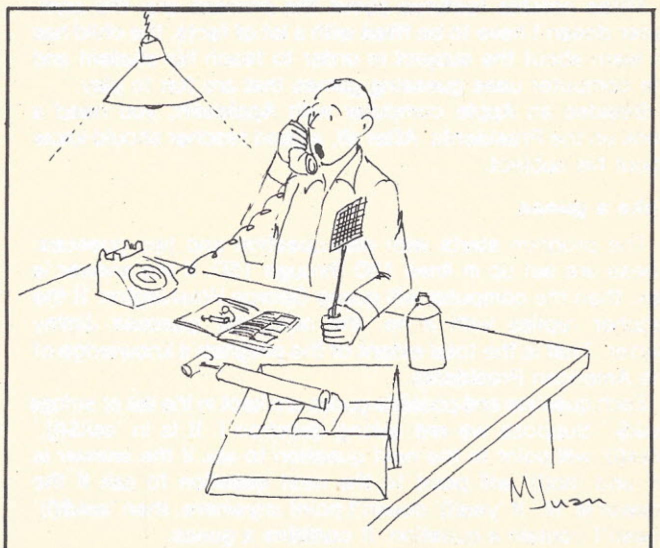
When the two courts allowed the software to be currently deducted in the year of its acquisition by the taxpayer in litigation, they allowed it under the Internal Revenue Code Section permitting "ordinary and necessary" business expense deduction. Among other factors, "ordinary and necessary" has been interpreted to mean the acquisition of an asset having a useful life not exceeding one year. The Idaho taxpayer, concluded the two courts, brought efficiency by purchasing the computer programs and held efficiency to be currently deductible.

Another recent case has suggested that, where a Board of Directors elected not to purchase a computer system to make their business more cost efficient where such a system was available, it may be liable to its shareholder-owners for improper business judgement. For many corporate directors and officers, this glimpse of futureshock may become a present reality. The point I want to make is that a company may be deemed negligent in not taking advantage of a more cost efficient means to operate.

To return to the Idaho case, I have thus far presented the extreme opposing positions: amortization vs. current deductibility. Is there a middle ground? Yes, argued one judge; he argued that the IRS already allows software write-offs more quickly than over their useful life by virtue of Internal Revenue Service Revenue Procedure 69-21 which permits a software acquirer to write-off his purchase over its useful life, 60 months, or shorter if the taxpayer can show that the software will have a useful life of less than 60 months in the taxpayer's trade or business. Strange, but why didn't the IRS choose to attack taxpayer's current deductibility argument by this revenue procedure? The same judge felt other judges had erred when they sided in favor of allowing current deductibility.

He compared the purchase of the software by the taxpayer as the same as the turn-of-the-century purchase of a player piano. I fail to comprehend this line of reasoning because a player piano made the tavern owner's business more profitable by eliminating the piano player's salary in addition to its novelty. Nevertheless, this judge failed to strike a responsive chord with his colleagues because he was the minority dissenting opinion. More importantly, this opinion points out the vast technological gap between present business reality vs. archaic judicial analogies.

To summarize, where a business purchases programs and associated operating manuals to establish or enhance its business services or products where immediate cost efficiency is not the taxpayer's primary objective, the chances of obtaining current deductibility for its purchase decrease. To date, there exists at least three major cases where software was permitted to be currently deducted. All fought and won against the positions taken by the IRS. The IRS will, I believe, continue to forum shop for another court which it forecasts will be more receptive to its capitalization argument. □



"Now, about this Debugging Kit you sent me..."

AL BAKER'S GAME CORNER

Computers As Teacher's Pets

Kids are natural teachers who love to show how smart they are. Listen to them. They're always explaining things to each other and their parents. What a child needs is a good, patient student.

Kids also love guessing games and so does an Apple computer. The Apple doesn't know much, but it learns fast and is never bored. . .it's a natural student.

Good teacher and good student—put them together and you have education. The Apple is learning from its young teacher and both are having fun.

But isn't it the young boy or girl who is supposed to be learning? If you think this is a problem, you've never taught. No one ever learns more than the teacher. The best way to get someone, even a child, to learn, is to force him to teach. Those students who surpass their teachers do so only because they have found a better teacher—themselves. They have learned the secret of learning. That's why children are natural teachers before they are taught to be students.

The Presidents

The basic concept behind this program has been around for years. Apple Computer has a similar program called Animals on its DOS 3.2 program disk.

The concept is simple, and the program is easy to understand. The child thinks up the name of a President of the United States. The computer tries to guess who it is. If the computer is wrong, the child tells the computer what additional question it should have asked. The young teacher has taught its electronic student a lesson in American history.

Three notable features about the process are: the computer doesn't have to be filled with a lot of facts, the child has to learn about the subject in order to teach his student and the computer uses guessing games that are fun to play.

Besides an Apple computer with Applesoft, you need a book on the Presidents. After all, a good teacher should know about his subject.

Take a guess

The program starts with one question and two guesses. These are set up in lines 130 through 150. If the answer is yes, then the computer will guess George Washington. If the teacher replies with a no, the computer guesses Jimmy Carter. That is the total extent of the program's knowledge of the American Presidents.

Each question and possible guess are kept in the list of strings 'ask$.' Suppose we are asking question I. It is in 'ask$(I).' 'Yes(I)' will point to the next question to ask if the answer is 'Y' and 'no(I)' will point to the next question to ask if the answer is 'N.' If 'yes(I)' doesn't point anywhere, then 'ask$(I)' doesn't contain a question. It contains a guess.

Lines 220 through 270 ask the questions. The program always starts with question zero. It continues asking questions until it encounters a question with a zero value in 'yes(I).' The program has run out of questions. It is time to guess.

Lines 310 to 340 make the guess and lines 380 to 400 handle a correct response. The remainder of the program handles an incorrect guess.

The variable 'empty' points to the next open slot in the 'ask$' array. Here we place the correct answer from our teacher. Next, we take the incorrect guess and remove it from 'ask$(I)' and place it in the next open slot in 'ask$.' The program will use the new question put in 'ask$(I)' to choose between the guesses in 'ask$(empty)' and 'ask$(empty + 1).' If the correct answer for the new guess is 'Y' then line 520 sets up the yes and no arrays. Otherwise line 530 does the work. Now the program is ready and able to try again.

The eager young teacher will shortly have a well-educated student. But don't stop there; this program is easily modified. How about teaching the computer about the states or animals, English novelists, or Bible characters. Almost any subject can be taught to the Apple. Who knows. . .you might end up with the best educated computer (and child) in the neighborhood. □

PROGRAM LISTING

```
10 REM      THE PRESIDENTS
20 REM
30 REM
40 REM
50 REM  ASK$ = QUESTIONS AND ANSWERS
60 REM  YES  = POINTER TO YES RESPONSE
70 REM  NO   = POINTER TO NO RESPONSE
80 REM   EMPTY= NEXT UNUSED ASK$ LOCATION
90 REM
100 CALL  - 936
110 DIM ASK$(100),YES(100),NO(100)
130 ASK$(0) = "DID HE TAKE OFFICE BEFORE 1876"
140 YES(0) = 1:ASK$(1) = "GEORGE WASHINGTON"
150 NO(0) = 2:ASK$(2) = "JIMMY CARTER"
160 EMPTY = 3
170 REM
180 REM  ASK QUESTIONS UNTIL I THINK I KNOW WHO IT IS
190 REM
200 PRINT "THINK OF A PRESIDENT OF THE UNITED"
210 PRINT "STATES.  I WILL TRY TO GUESS WHO IT IS "
220 I = 0
230  IF YES(I) = 0 GOTO 310
240  PRINT ASK$(I);: INPUT " (Y,N)?";ANS$
250  IF ANS$ = "Y" THEN I = YES(I): GOTO 230
260  IF ANS$ = "N" THEN I = NO(I): GOTO 230
270  GOTO 240
280  REM
290  REM  I THINK I KNOW
300  REM
310  PRINT "IS IT ";ASK$(I);: INPUT " (Y,N)?";ANS$
320  IF ANS$ = "N" GOTO 440
```

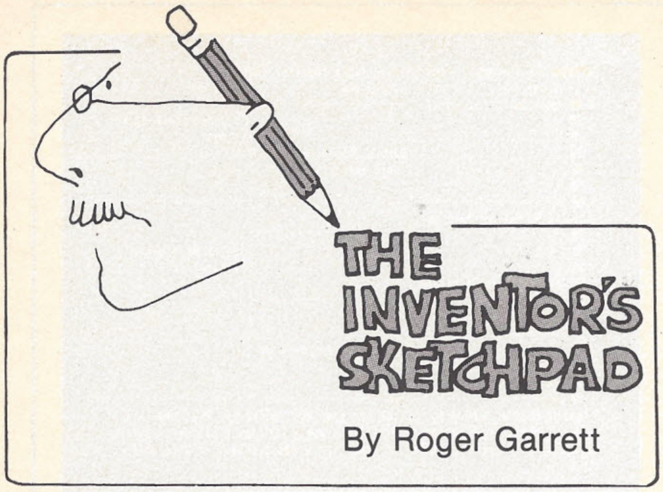
```
330 IF ANS$ = "Y" GOTO 380
340 GOTO 310
350 REM
360 REM  I GUESSED HIM
370 REM
380 PRINT "YEPEEEEE. THAT WAS FUN, I WANT TO"
390 PRINT "GUESS ANOTHER ONE."
400 GOTO 200
410 REM
420 REM  I GUESSED WRONG
430 REM
440 PRINT "OOPS. YOU GOT ME."
450 INPUT "WHO WAS IT?";ASK$(EMPTY)
460 ASK$(EMPTY + 1) = ASK$(I)
470 PRINT "I SHOULDN'T HAVE GUESSED SO QUICKLY."
480 PRINT "WHAT QUESTION SHOULD I HAVE ASKED?"
490 PRINT "(REMEMBER THAT THE QUESTION MUST HAVE"
500 PRINT " A YES OR NO ANSWER.)"
505 INPUT ASK$(I)
510 INPUT "IS THE CORRECT ANSWER YES OR NO (Y,N)?";ANS$
520 IF ANS$ = "Y" THEN YES(I) = EMPTY:NO(I) = EMPTY + 1: GOTO 550
530 IF ANS$ = "N" THEN NO(I) = EMPTY:YES(I) = EMPTY + 1: GOTO 550
540 GOTO 510
550 PRINT "MAYBE I'LL DO BETTER THIS TIME, SO LET"
560 PRINT "ME TRY AGAIN"
570 EMPTY = EMPTY + 2
580 GOTO 200
```

SAMPLE OUTPUT

```
THINK OF A PRESIDENT OF THE UNITED
STATES. I WILL TRY TO GUESS WHO IT IS.
DID HE TAKE OFFICE BEFORE 1876 (Y,N)?N
IS IT JIMMY CARTER (Y,N)?N
OOPS. YOU GOT ME.
WHO WAS IT?JOHN KENNEDY
I SHOULDN'T HAVE GUESSED SO QUICKLY.
WHAT QUESTION SHOULD I HAVE ASKED?
(REMEMBER THAT THE QUESTION MUST HAVE
 A YES OR NO ANSWER.)
?DID HE DIE IN OFFICE
IS THE CORRECT ANSWER YES OR NO (Y,N)?Y
MAYBE I'LL DO BETTER THIS TIME, SO LET
ME TRY AGAIN
THINK OF A PRESIDENT OF THE UNITED
STATES. I WILL TRY TO GUESS WHO IT IS.
DID HE TAKE OFFICE BEFORE 1876 (Y,N)?N
DID HE DIE IN OFFICE (Y,N)?Y
IS IT JOHN KENNEDY (Y,N)?Y
YEPEEEEE. THAT WAS FUN, I WANT TO
```

```
GUESS ANOTHER ONE.
THINK OF A PRESIDENT OF THE UNITED
STATES. I WILL TRY TO GUESS WHO IT IS.
DID HE TAKE OFFICE BEFORE 1876 (Y,N)?N
DID HE DIE IN OFFICE (Y,N)?Y
IS IT JOHN KENNEDY (Y,N)?N
OOPS. YOU GOT ME.
WHO WAS IT?FRANKLIN ROOSEVELT
I SHOULDN'T HAVE GUESSED SO QUICKLY.
WHAT QUESTION SHOULD I HAVE ASKED?
(REMEMBER THAT THE QUESTION MUST HAVE
 A YES OR NO ANSWER.)
?DID HE LEAD DURING A MAJOR WAR
IS THE CORRECT ANSWER YES OR NO (Y,N)?Y
MAYBE I'LL DO BETTER THIS TIME, SO LET
ME TRY AGAIN
THINK OF A PRESIDENT OF THE UNITED
STATES. I WILL TRY TO GUESS WHO IT IS.
DID HE TAKE OFFICE BEFORE 1876 (Y,N)?N
DID HE DIE IN OFFICE (Y,N)?N
IS IT JIMMY CARTER (Y,N)?N
OOPS. YOU GOT ME.
WHO WAS IT?DWIGHT EISENHOWER
I SHOULDN'T HAVE GUESSED SO QUICKLY.
WHAT QUESTION SHOULD I HAVE ASKED?
(REMEMBER THAT THE QUESTION MUST HAVE
 A YES OR NO ANSWER.)
?WAS HE A FAMOUS GENERAL
IS THE CORRECT ANSWER YES OR NO (Y,N)?Y
MAYBE I'LL DO BETTER THIS TIME, SO LET
ME TRY AGAIN
THINK OF A PRESIDENT OF THE UNITED
STATES. I WILL TRY TO GUESS WHO IT IS.
DID HE TAKE OFFICE BEFORE 1876 (Y,N)?N
DID HE DIE IN OFFICE (Y,N)?Y
DID HE LEAD DURING A MAJOR WAR (Y,N)?Y
IS IT FRANKLIN ROOSEVELT (Y,N)?Y
YEPEEEEE. THAT WAS FUN, I WANT TO
GUESS ANOTHER ONE.
THINK OF A PRESIDENT OF THE UNITED
STATES. I WILL TRY TO GUESS WHO IT IS.
DID HE TAKE OFFICE BEFORE 1876 (Y,N)?N
DID HE DIE IN OFFICE (Y,N)?N
WAS HE A FAMOUS GENERAL (Y,N)?Y
IS IT DWIGHT EISENHOWER (Y,N)?Y
YEPEEEEE. THAT WAS FUN, I WANT TO
GUESS ANOTHER ONE.
THINK OF A PRESIDENT OF THE UNITED
STATES. I WILL TRY TO GUESS WHO IT IS.
```


CIRCLE INQUIRY NO. 46



By Roger Garrett

Building a Simulation System

"Computer!"

"Yes?"

"Undersea exploration mission, please."

"Certainly."

My standard keyboard and display dissolve into blackness and a minisub control console materializes in its place. An array of dials and gauges surround a color radar/status control display directly in front of me. Two joysticks, for directional and speed control, appear at the ends of my armrests. Behind me, where nothing existed before, I hear the whir of the craft's engines and the sounds of my assistant, Jack, puttering around at his console preparing for our mission.

"Do you wish a standard checkout run or shall I present a task to you?"

I have already run 13 checkout runs this month and am satisfied that all systems are in full functional order. More importantly, I am confident of my own skills in operating this experimental minisub.

"Select a task. A difficult one."

Perhaps I shouldn't be so bold. Confidence occasionally gets the better of reason.

"As you wish."

"Your position is the Gulf of Mexico. Depth, 30 meters. Seas rough. Two kilometers north-northeast an oil well has blown. Three men are missing."

Not quite that difficult, I think to myself. But I accept the challenge.

I make a visual check of the control panel. Everything nominal. "Everything ok with you?" I call to Jack.

"Just fine here."

"Then let's go."

I push forward on my right joystick. The engines speed up. With a slight twisting motion of my left hand, the minisub begins turning, slowly, to the correct heading; north-northeast. Full forward on joystick one and we are cruising at 60 knots towards the crippled oil well.

"Computer!"

"Yes?"

"Everything is working fine but I really could use an overhead display, and perhaps one on either side of my control station, to give me views of the ocean. Something like windows. It's a little claustrophobic with this one little viewport."

"How's this?"

The additional viewports appear as requested. The one above shows the ocean surface, choppy but not dangerous. To my left and right I can see the midnight blackness of the sea. This plastic ocean apparently contains no life. Not as realistic as I would like but I can change that next time around.

"Just fine, thanks."

Disaster strikes

We finally arrive at the oil well and I begin to regret my request for a challenging mission. I can't see what's happening on the surface but down here it looks like a Kansas tornado.

The drilling rig has broken right at the base and the plume of escaping oil, rising rapidly to the surface, has set up a vortex threatening to pull us into it. Emergency reverse thrust slows our advance towards the rig, but is not sufficient to completely pull us away. I should have requested more information from topside before getting so close, but that's just a judgement error. What matters is that the sub can't resist the force of the current. I could ask the computer to modify the engine design but at this point that would be cheating. I could take care of that later. For now I push the sub to its limit and take my chances.

I ask Jack for a status check of the explosives deployment system. "Everything nominal," he responds. "Nominal," I repeat to myself. I've got to give that guy more vocabulary.

I change thrust to half-forward and aim the sub at the base of the rig. The vortex quickly takes hold and we spiral inward, being swept up with debris from the damaged rig, sand, and a stray shrimp (there are fish after all).

The sub shudders against the strain as I try to keep us headed directly at the base of the rig. It is no easy task. We are now circling at one revolution every 45 seconds. The view is beginning to blur. Too much sand, too much oil. And my inner ears are screaming for relief. I have to take the chance that I am finally close enough.

"Deploy!" I yell back to Jack. ("Deploy?" I think to myself. "Fire" or "bombs away" would make more sense. But I had fallen into the mundane vocabulary of my assistant.)

The explosive canister shoots out from the deployment tube directly towards the base of the billowing oil. But the vortex is stronger than the canister's propulsion system. It spirals faster and faster into the very heart of the black cone. A steel beam, jutting out from what is left of the rig, catches the canister with full force.

Did we make it?

Jack vanishes. The control panels blink once, then twice, then dissolve away. In their place my keyboard and CRT once again appear.

"Simulation over." The voice of the computer finally breaks the silence.

I sit motionless for several minutes taking deep breaths to regain what little composure I can before responding.

"How did I do?", I ask.

"You stopped the oil spill. . ."

"Great!", I interrupt.

"But you and your assistant were killed. Too close to the explosion. The minisub just can't take that kind of force."

"Damn," I think to myself, "I was sure I was far enough away."

"And you apparently forgot about the three missing men."

I had, but how many things can two men (well, one man and one "assistant") accomplish in such a dangerous situation? I was about to argue my point but reconsidered. The computer just wouldn't understand.

I reach forward and flick the abort switch on my keyboard which then dissolves away to join the rest of my imaginary hardware, safely tucked away in the memory of my simulation computer. Overhead lights flash on, revealing the four walls of my evaluation/training cubicle. I get up from my chair and carefully remove the telepresence suit, leaving it to dangle from the myriad of position control units suspended from the ceiling.

I begin to leave the room but pause momentarily at the door. "Tomorrow," I say quietly, "I'll save that oil well *and* the three missing men."

Back to reality

By now, if you are a regular reader of this column, you are probably wondering whether the editor goofed. Or maybe you think I've gone off the deep end. After all, can I possibly imply that a simulation system as sophisticated as the one de-

scribed be built? Yes, of course. Do I really think that it can be done using personal computers? Yes again. Do I think that you can build one? Well, maybe. It may not be as fancy, but it is certainly worth experimenting with.

In the June 1980 issue of *Omni* magazine, Dr. Marvin Minsky wrote about a concept which he calls "telepresence." In simple terms it means "existing elsewhere." His article describes how, with suitable hardware, you can be made to "feel" as though you are no longer where you are, but rather where you want to be.

There are three basic components to such a system. First is a robot with arms, hands, fingers, a head with eyes and ears, legs, feet and toes.

The second is a special set of hardware which the operator wears, similar to a suit. Within is an abundance of position sensors which, when read, can tell the positional status of every part of your body. It also has movement effectors which can impart forces to your body causing, for example, your hand to form a fist. Other effectors can cause heat or texture sensations. Mounted inside the helmet is a stereo headphone set to provide audio information, a stereo CRT set to provide visual information and a microphone to issue commands or communicate with others.

And finally there is a controlling computer linked to both your suit and the robot, and which oversees the entire operation. Everything you do or experience is transferred via the computer to the robot. Everything that the robot does or experiences is transferred back to you. When you raise your arm the robot raises its arm. What the robot "sees" via its TV camera "eyes," you see via your stereo CRTs. When you look to the right you see not what is to your right but what is to the robot's right. Your very presence is transferred to the robot, whether it be in the next room or on the moon. You experience telepresence.

As Dr. Minsky explains, such devices would be invaluable. They could perform all of the life-threatening jobs such as coal mining, nuclear reactor maintenance, undersea exploration, space station construction. Humans would work in the air conditioned comfort of a work station while the robot did the dirty and dangerous work.

A simulated existence

What I have in mind is artificial telepresence. We do away with the robot completely, leaving only the operator's telepresence suit (OTS) and the controlling computer. All sensations, sounds and images which the operator experiences would be completely computer generated. We transfer the operator's presence not into a hulk of robotic hardware but into a simulated environment which "exists" only within the software of the computer.

Now what would we use the aritificial telepresence system (ATS) for? Suppose we are designing an aircraft. Instead of simple engineering drawings, the ATS can construct a 3-D image of the plane which the operator can walk around and touch. He can sit in the pilot's seat to verify that there is suitable visibility out the windows. He can make sure that all the controls and displays are in proper positions. There is no need to build a hardware prototype since the ATS can build it in software. Design changes can be made simply and quickly with no retooling required.

How about astronaut training? Rather than building one-of-a-kind space shuttle simulators, we simulate the hardware within the software of the ATS, strap the astronaut candidate into a telepresence suit, and he can "fly" missions as simple or as complex as he desires.

What I have introduced is the concept of virtual hardware. The system provides the user with the ability to touch, to feel, to manipulate, to experience hardware that does not actually exist. Just as we can access vast quantities of computer memory that does not actually exist within a computer via virtually memory addressing, we can utilize hardware that is only imaginary via this virtual hardware concept. Obviously

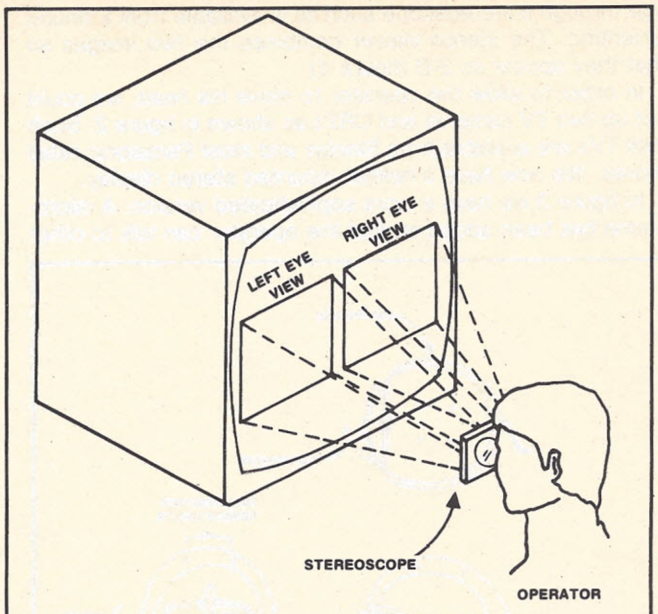


Figure 1. Presenting 3-D illusions.

this concept provides for a wide range of applications. But how do we go about building such a device?

The most important feature of the ATS is the ability to present 3-D illusions. Assuming that the computer contains a description of a set of 3-D objects, it can draw a projected 2-D image of those objects on a CRT. (A software package such as Appleworld by United Software of America in NY, NY for the Apple computer provides such a capability.)

This gives the impression of dimensionality but does not provide the depth perception which we obtain from our stereo vision. To provide this we could generate two separate images for each eye on the screen, with each presenting a slightly different perspective, and then view the im-

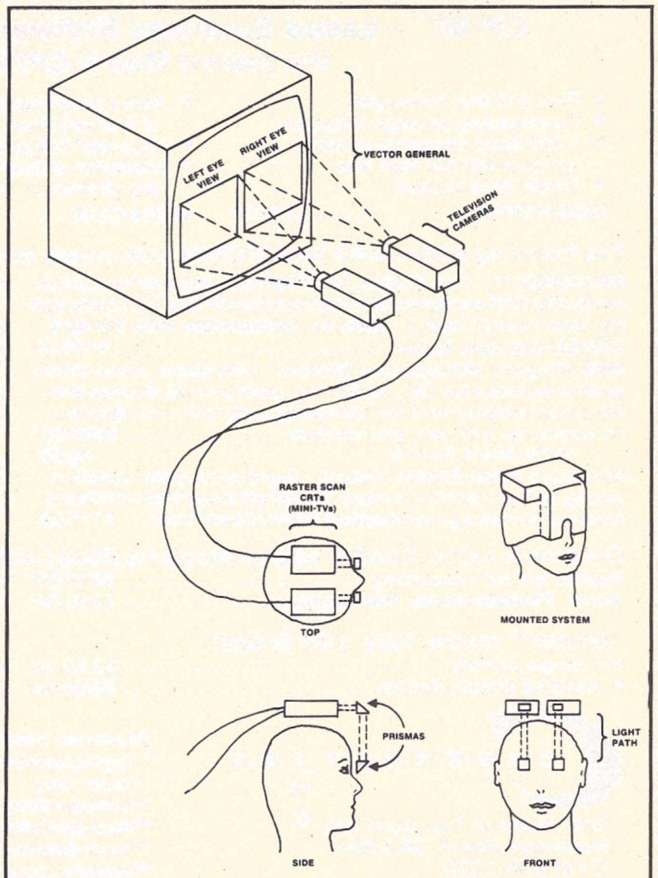


Figure 2. Helmet mounted stereo display.

age through a stereoscope such as is available from Edmund Scientific. The stereo viewer combines the two images so that they appear as 3-D (figure 1).

In order to allow the operator to move his head, we could set up two TV cameras and CRTs as shown in figure 2. Such mini TVs are available from Sinclair and most Panasonic retail stores. We now have a helmet-mounted stereo display.

In figure 3 we have a more sophisticated version. A microphone has been added so that the operator can talk to other

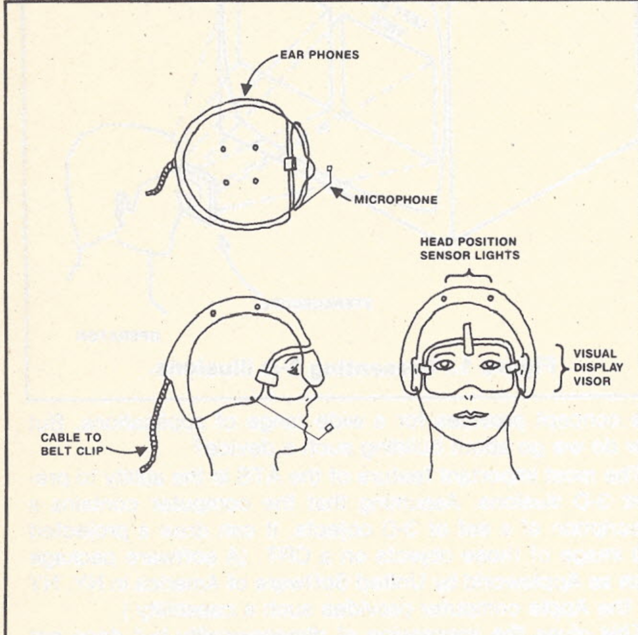


Figure 3. Microphone added to the helmet.

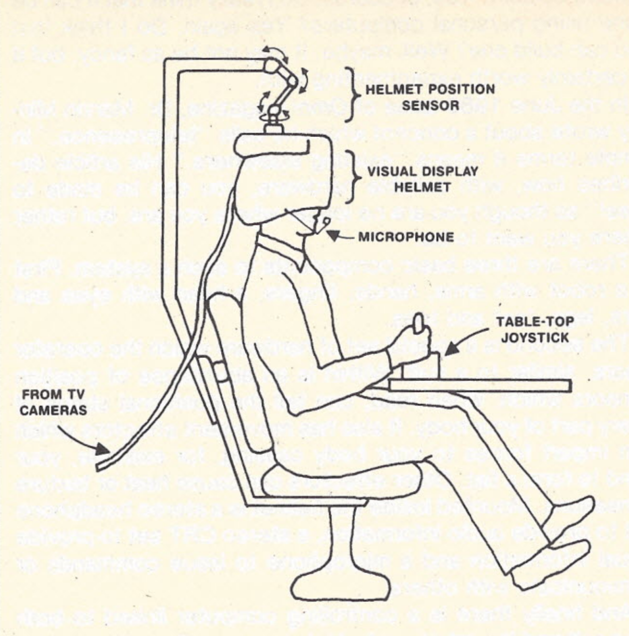


Figure 4. Operator sits in chair with helmet connected to a multi-jointed arm system.

operators or to the computer. The helmet contains a set of stereo earphones so that spatially encoded audio information can be presented to him. (IA, Dec 1979, pg. 46)

Now consider what happens when the operator is viewing the inside of a minisub and he turns his head to the right. Logically we would assume that he sees the right inside of the sub. But with our helmet-mounted display, thus far described, and the computer generating the image, the com-

puter does not know the operator has turned his head, implying that a modified image must now be displayed.

So, as shown in figure 4, we position the operator in a chair and connect his helmet to a multi-jointed arm system. Each angular and rotational joint is connected to a variable resistor which the computer can read. The computer then can know the exact position and direction of the operator's head and use this information as it updates the displayed image. Whatever position the operator turns his head the image changes and he experiences the computer-generated world of images as though he were really inside it.

Reaching for objects

Now that our operator feels as though he is inside the virtual environment, he needs some method of interacting with it rather than merely viewing it. In figure 4 he has a joystick. Perhaps he could also use a keyboard, but manipulating objects with these devices is not man-machine efficient. What

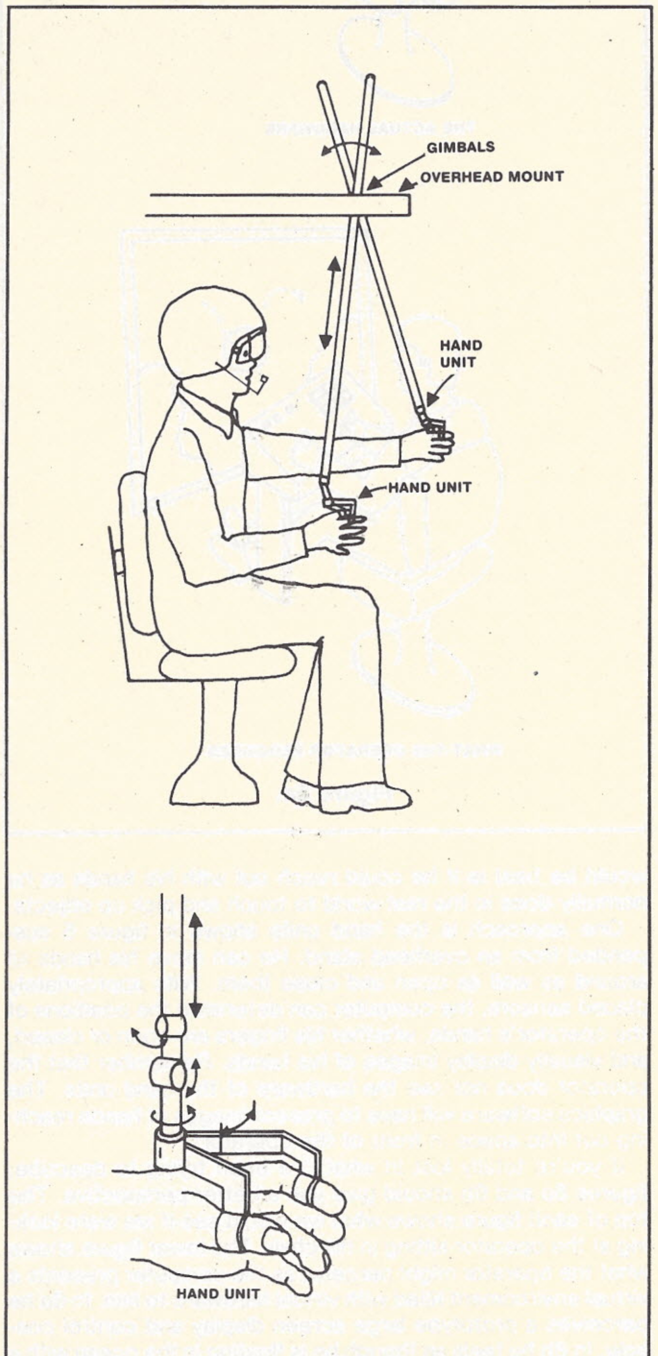
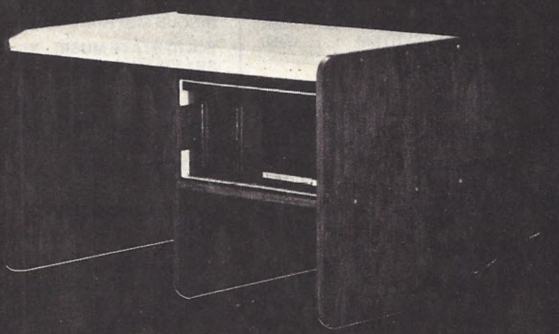
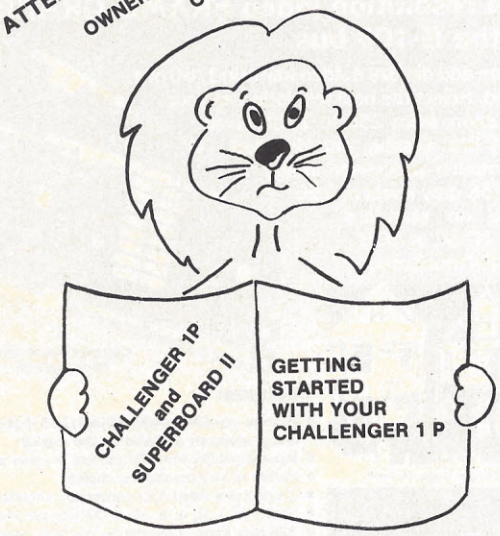
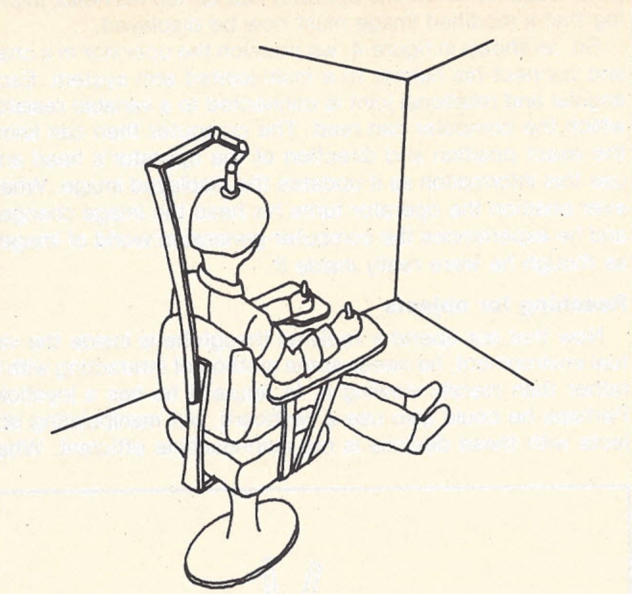


Figure 5. Manipulating objects with hand units.







THE ACTUAL HARDWARE

WHAT THE OPERATOR PERCEIVES

Figure 6a.

would be best is if he could reach out with his hands as he normally does in the real world to touch and pick up objects.

One approach is the hand units shown in figure 5 suspended from an overhead stand. He can move his hands all around as well as open and close them. With appropriately placed sensors, the computer can determine the positions of the operator's hands, whether his fingers are open or closed, and visually display images of his hands. Remember that the operator does not see the hardware of the hand units. The graphics software will have to present images of hands reaching out into space in front of the operator.

If you're totally lost in what I've been trying to describe, figures 6a and 6b should give you a better perspective. The top of each figure shows what we would see if we were looking at the operator sitting in his chair. The lower figure shows what the operator might perceive as the computer presents a virtual environment filled with virtual hardware to him. In 6a he perceives a prototype large screen display and control console. In 6b he feels as though he is floating in the ocean with a submarine passing several kilometers away.

As an alternative to overhead-mounted arm position sensors, we might attach sensors directly to the operator's arms,

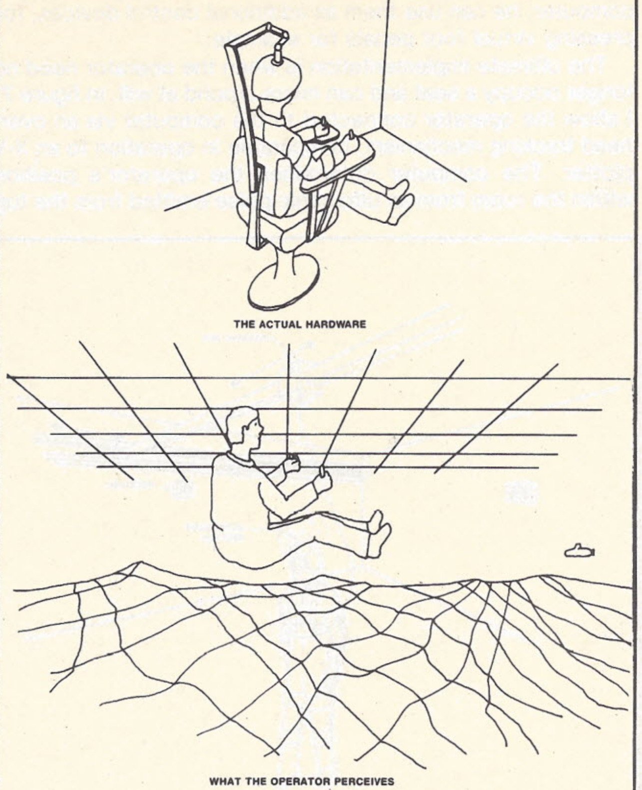


Figure 6b.

hands, legs and feet. This would give him greater flexibility and, since his leg and foot positions are now known by the computer, he can use them as additional control devices, for pressing virtual foot pedals for example.

The ultimate implementation is when the operator need no longer occupy a seat and can move around at will. In figure 7, I show the operator connected to the computer via an overhead tracking mechanism, comparable in operation to an X-Y plotter. The computer determines the operator's position within the room from an ultrasonic pulse emitted from the top

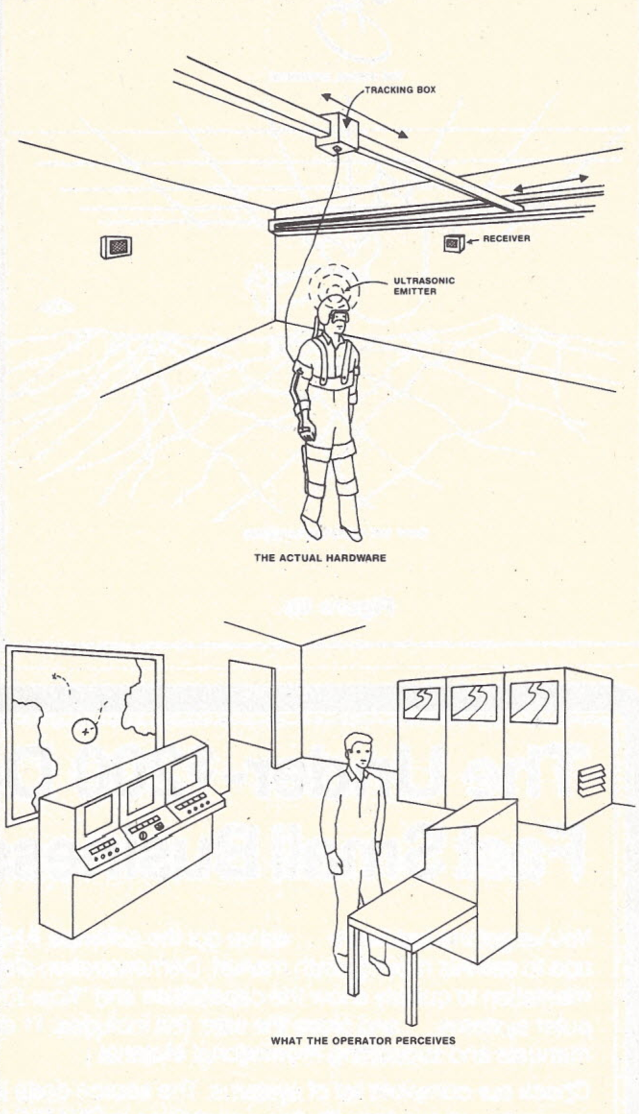
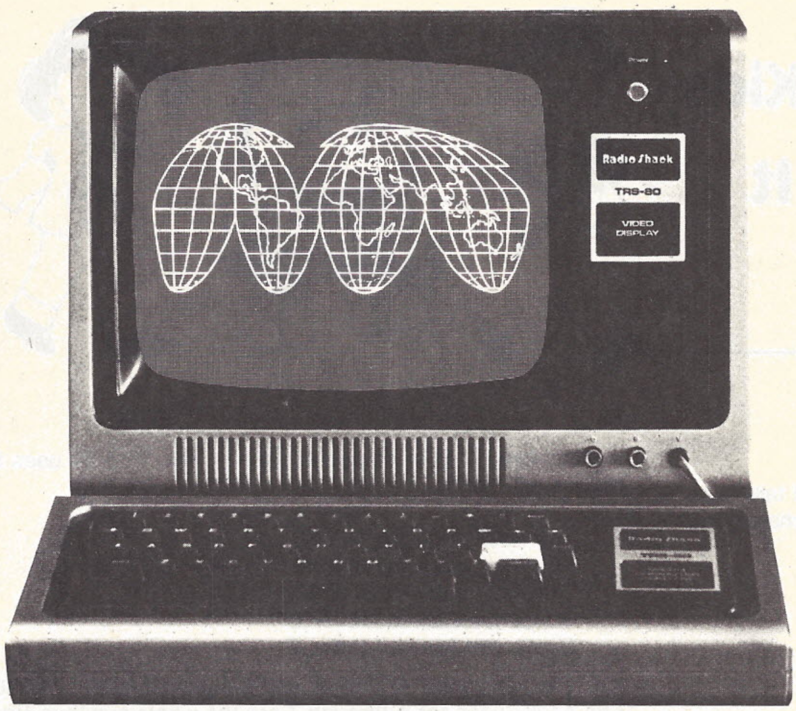


Figure 7. The operator connected tracking machine.

of the operator's helmet and detected by receivers mounted in the walls.

An alternative method would be to have the helmet directly connected to the overhead tracking box in a manner similar to that shown in figure 4.

Virtually any type of environment could be simulated. Here I have shown that a prototype command control system is being designed by the operator. He can move the pieces of "hardware" simply by reaching out and pushing it. The system provides force feedback so he actually feels as though he is pushing a physical object while in reality he is only manipulating a database item within the simulation computer. He can reach out and press a virtual button and watch the result on a virtual flat-panel display. Of course, if he tries to walk through that virtual door he is going to quickly appreciate the difference between reality and fantasy. □



CompuServe's information service.

A world of information available. Right now.

If you have a personal computer—or a computer terminal — CompuServe can bring a world of information into your home or small business.

CompuServe
CompuServe is a major computer services company that has been serving top industrial and governmental clients for more than a decade. Now we're also applying our extensive computing capacity to the delivery of information services to your personal computer.

The Information Service
This exciting service is available in more than 250 cities between 6 p.m. and 5 a.m. weekdays, all day weekends and most holidays. Cost? Only $5.00 per hour billed in minutes. All you need is a 300 baud modem, and our complete information service is only a local phone call away.

Welcome to CompuServe's information service.

- *News. Weather. Sports.* Major regional newspapers. Plus international news services.
- *Finance. MicroQuote.* Updates and historical information on stocks, bonds and commodities.
- *Entertainment.* Theatre, book, movie and restaurant reviews. Plus opera, symphony, ballet, dance, museums, galleries...
- *Electronic Mail.* Create, edit, send and receive messages from any other CompuServe user... nationwide.
- *Home & Educational Reference Service.* Anything you want to know... from encyclopedia information to household tips.
- *CompuServe User Information.* In case you need technical help... and information on new services as they become available.
- *MicroNET.* For the computer hobbyist. Software Exchange, line printer art gallery, challenging games, programming languages, word processing, business & educational programs... and much, much more.

And this is just the beginning. CompuServe is continually adding to its list of available services. The world of electronic information isn't coming tomorrow. It's here today.

CompuServe is Available at Radio Shack® Stores
Your local Radio Shack® Store Sells CompuServe's information service. There are more than 6,000 Radio Shack® Stores and Dealers nationwide. Check with the outlet nearest you.

Radio Shack is a division of Tandy Corporation.

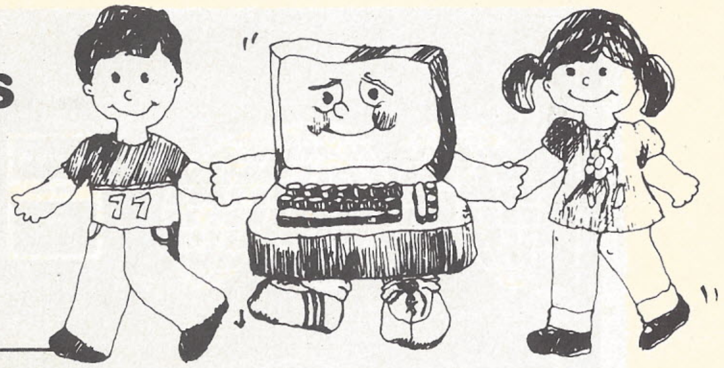
CompuServe

Information Service Division
5000 Arlington Centre Blvd.
Columbus, Ohio 43220
(614) 457-8600

My TRS-80 Likes Me

When I Teach Kids How to Use It

by Bob Albrecht



Ascii Land

Wandering Star is on vacation. This time we will talk more about those numeric codes that stand for characters. Remember:

Wandering Star (*) is 42.
Cosmic dust (.) is 46.
Empty space (" ") is 32.

Every keyboard character has its own numeric code, called an Ascii. Ascii means American Standard Code for Information Interchange. Lots of computers use it. The TRS-80 uses it. So, we will use it.

The Ascii code for A is 65.
The Ascii code for B is 66.
The Ascii code for C is 67.
Guess the Ascii code for D.
Guess the Ascii code for Z.

Yes, the Ascii code for D is 68 and the Ascii code for Z is 90. The Ascii codes for A through Z are the whole numbers from 65 to 90.

Basic provides a built-in function called ASC which gives the Ascii code for any character. Clear the screen and try these.

```
You type:   PRINT ASC("A")
It prints:  65
You type:   PRINT ASC("B")
It prints:  66
You type:   PRINT ASC("Z")
It prints:  90
You type:   PRINT ASC("*")
It prints:  42
You type:   PRINT ASC(" ")
It prints:  32
```

Experiment! Use the following program to find the Ascii codes for keyboard characters.

```
100 REM**ASCII CODES FOR KEYBOARD CHARACTERS
110 CLS
120 PRINT : INPUT "KEYBOARD CHARACTER" ; KEY$
130 CODE = ASC(KEY$)
140 PRINT "THE ASCII CODE IS " CODE
150 GOTO 120
```

Here is what happened when we ran the program.

```
KEYBOARD CHARACTER?    A
THE ASCII CODE IS      65

KEYBOARD CHARACTER?    *
THE ASCII CODE IS      42

KEYBOARD CHARACTER?    1
THE ASCII CODE IS      49

KEYBOARD CHARACTER?    " "
THE ASCII CODE IS      32
```

Try some keys yourself. For most keys, the program will work. If your exploration is quite thorough, you may run into

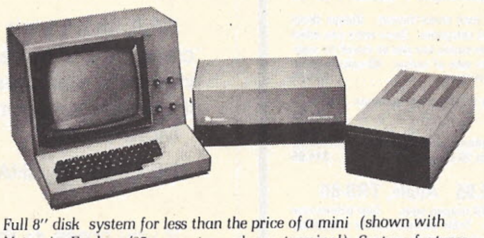
Start learning and computing for only ***$129.95*** *with a Netronics 8085-based computer kit. Then expand it in low-cost steps to a business/development system with 64k or more RAM, 8" floppy disk drives, hard disks and multi-terminal I/O.*

THE NEW EXPLORER/85 SYSTEM

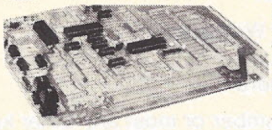
Special! *Full 8" floppy, 64k system for less than the price of a mini! Only* **$1499.95!**

(Also available wired & tested, $1799.95)

Imagine — for only $129.95 you can own the starting level of Explorer/85, a computer that's expandable into full business/development capabilities — a computer that can be your beginner system, an OEM controller, or an IBM-formatted 8" disk small business system. From the first day you own Explorer/85, you begin computing on a significant level, and applying principles discussed in leading computer magazines. Explorer/85 features the advanced Intel 8085 cpu, which is 100% compatible with the older 8080A. It offers onboard S-100 bus expansion, Microsoft BASIC in ROM, plus instant conversion to mass storage disk memory with standard IBM-formatted 8" disks. All for only $129.95, plus the cost of power supply, keyboard/terminal and RF modulator if you don't have them (see our remarkable prices below for these and other accessories). With a Hex Keypad/display front panel, Level "A" can be programmed with no need for a terminal, ideal for a controller, OEM, or a real low-cost start.



Full 8" disk system for less than the price of a mini (shown with Netronics Explorer/85 computer and new terminal). System features floppy drive from Control Data Corp., world's largest maker of memory storage systems (not a hobby brand!)



Level "A" is a complete operating system, perfect for beginners, hobbyists, industrial controller use. $129.95

Level "A" With Hex Keypad/Display.

LEVEL "A" SPECIFICATIONS

Explorer/85's Level "A" system features the advanced Intel 8085 cpu, an 8355 ROM with 2k deluxe monitor/operating system, and an advanced 8155 RAM I/O . . . all on a single motherboard with room for RAM/ROM/PROM/EPROM and S-100 expansion, plus generous prototyping space.

PC Board: Glass epoxy, plated through holes with solder mask. • **I/O:** Provisions for 25-pin (DB25) connector for terminal serial I/O, which can also support a paper tape reader . . . cassette tape recorder input and output . . . cassette tape control output . . . LED output indicator on SOD (serial output) line . . . printer interface (less drivers) . . . total of four 8-bit plus one 6-bit I/O ports. • **Crystal Frequency:** 6.144 MHz. • **Control Switches:** Reset and user (RST 7.5) interrupt . . . additional provisions for RST 5.5, 6.5 and TRAP interrupts onboard. • **Counter/Timer:** Programmable, 14-bit binary. • **System RAM:** 256 bytes located at F800, ideal for smaller systems and for use as an isolated stack area in expanded systems . . . RAM expandable to 64K via S-100 bus or 4k on motherboard.

System Monitor (Terminal Version): 2k bytes of deluxe system monitor ROM located at FØØØ, leaving ØØØØ free for user RAM/ROM. Features include tape load with labeling . . . examine/change contents of memory . . . insert data . . . warm start . . . examine and change all registers . . . single step with register display at each break point, a debugging/training feature . . . go to execution address . . . move blocks of memory from one location to another . . . fill blocks of memory with a constant . . . display blocks of memory . . . automatic baud rate selection to 9600 baud . . . variable display line length control (1-255 characters/line) . . . channelized I/O monitor routine with 8-bit parallel output for high-speed printer . . . serial console in and console out channel so that monitor can communicate with I/O ports.

System Monitor (Hex Keypad/Display Version): Tape load with labeling . . . tape dump with labeling . . . examine/change contents of memory . . . insert data . . . warm start . . . examine and change all registers . . . single step with register display at each break point . . . go to execution address. Level "A" in this version makes a perfect controller for industrial applications, and is programmed using the Netronics Hex Keypad/Display. It is low cost, perfect for beginners.

HEX KEYPAD/DISPLAY SPECIFICATIONS

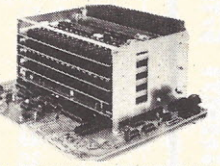
Calculator type keypad with 24 system-defined and 16 user-defined keys. Six digit calculator-type display, that displays full address plus data as well as register and status information.

LEVEL "B" SPECIFICATIONS

Level "B" provides the S-100 signals plus buffers/drivers to support up to six S-100 bus boards, and includes: address decoding for onboard 4k RAM expansion selectable in 4k blocks . . . address decoding for onboard 8k EPROM expansion selectable in 8k blocks . . . address and data bus drivers for onboard expansion . . . wait state generator (jumper selectable), to allow the use of slower memories . . . two separate 5 volt regulators.

LEVEL "C" SPECIFICATIONS

Level "C" expands Explorer/85's motherboard with a card cage, allowing you to plug up to six S-100 cards directly into the motherboard. Both cage and card are neatly contained inside Explorer's deluxe steel cabinet. Level "C" includes a sheet metal superstructure, a 5-card, gold plated S-100 extension PC board that plugs into the motherboard. Just add required number of S-100 connectors.



Explorer/85 With Level "C" Card Cage.

LEVEL "D" SPECIFICATIONS

Level "D" provides 4k of RAM, power supply regulation, filtering decoupling components and sockets to expand your Explorer/85 memory to 4k (plus the original 256 bytes located in the 8155A). The static RAM can be located anywhere from ØØØØ to EFFF in 4k blocks.

LEVEL "E" SPECIFICATIONS

Level "E" adds sockets for 8k of EPROM to use the popular Intel 2716 or the TI 2516. It includes all sockets, power supply regulator, heat sink, filtering and decoupling components. Sockets may also be used for 2k x 8 RAM IC's (allowing for up to 12k of onboard RAM).

DISK DRIVE SPECIFICATIONS

- 8" CONTROL DATA CORP. professional drive.
- LSI controller.
- Write protect.
- Single or double density.
- Data capacity: 401,016 bytes (SD), 802,032 bytes (DD), unformatted.
- Access time: 25ms (one track).

DISK CONTROLLER/ I/O BOARD SPECIFICATIONS

- Controls up to four 8" drives.
- 1771A LSI (SD) floppy disk controller.
- Onboard data separator (IBM compatible).
- 2 Serial I/O ports
- Autoboot to disk system when system reset.
- 2716 PROM socket included for use in custom applications.
- Onboard crystal controlled.
- Onboard I/O baud rate generators to 9600 baud.
- Double-sided PC board (glass epoxy.)

DISK DRIVE CABINET/POWER SUPPLY

- Deluxe steel cabinet with individual power supply for maximum reliability and stability.

ORDER A COORDINATED EXPLORER/85 APPLICATIONS PAK!

Beginner's Pak (Save $26.00!) — Buy Level "A" (Terminal Version) with Monitor Source Listing and AP-1 5-amp Power Supply: (regular price $199.95), now at SPECIAL PRICE: **$169.95** plus post. & insur.

Experimenter's Pak II (Save $53.40!) — Buy Level "A" (Hex Keypad/Display Version) with Hex Keypad/Display, Intel 8085 User Manual, Level "A" Hex Monitor Source Listing, and AP-1 5-amp Power Supply: (regular price $279.35), all at SPECIAL PRICE: **$219.95** plus post. & insur.

Special Microsoft BASIC Pak (Save $103.00!) — Includes Level "A" (Terminal Version), Level "B", Level "D" (4k RAM), Level "E", 8k Microsoft in ROM, Intel 8085 User Manual, Level "A" Monitor Source Listing, and AP-1 5-amp Power Supply: (regular price $439.70), now yours at SPECIAL PRICE: **$329.95** plus post. & insur.

ADD A TERMINAL WITH CABINET, GET A FREE RF MODULATOR: Save over $114 at this SPECIAL PRICE: **$499.95** plus post. & insur.

Special 8" Disk Edition Explorer/85 (Save over $104!) — Includes disk-version Level "A", Level "B", two S-100 connectors and brackets, disk controller, 64k RAM, AP-1 5-amp power supply, Explorer/85 deluxe steel cabinet, cabinet fan, 8" SD/DD disk drive from famous CONTROL DATA CORP. (not a hobby brand!), drive cabinet with power supply, and drive cable set-up for two drives. This package includes everything but terminal and printers (see coupon for them). Regular price $1630.30, all yours in kit at SPECIAL PRICE: **$1499.95** plus post. & insur. Wired and tested, only **$1799.95.**

Special! Complete Business Software Pak (Save $625.00!) — Includes CP/M 2.0, Microsoft BASIC, General Ledger, Accounts Receivable, Accounts Payable, Payroll Package: (regular price $1325), yours now at SPECIAL PRICE: **$699.95.**

Please send the items checked below:

- ☐ **Explorer/85 Level "A" kit** (Terminal Version) . . . **$129.95** plus $3 post. & insur.
- ☐ **Explorer/85 Level "A" kit** (Hex Keypad/Display Version) . . . **$129.95** plus $3 post. & insur.
- ☐ **8k Microsoft BASIC** on cassette tape. **$64.95** postpaid.
- ☐ **8k Microsoft BASIC in ROM kit** (requires Levels "B", "D" and "E") . . . **$99.95** plus $2 post. & insur.
- ☐ **Level "B" (S-100) kit** . . . **$49.95** plus $2 post. & insur.
- ☐ **Level "C" (S-100 6-card expander) kit** . . . **$39.95** plus $2 post. & insur.
- ☐ **Level "D" (4k RAM) kit** . . . **$69.95** plus $2 post. & insur.
- ☐ **Level "E" (EPROM/ROM) kit** . . . **$5.95** plus 50¢ p&h.
- ☐ **Deluxe Steel Cabinet** for Explorer/85 . . . **$49.95** plus $3 post. & insur.
  - ☐ Fan For Cabinet . . . **$15.00** plus $1.50 post. & insur.
- ☐ **ASCII Keyboard/Computer Terminal kit:** features a full 128 character set, u&l case; full cursor control; 75 ohm video output; convertible to baudot output; selectable baud rate, RS232-C or 20 ma. I/O, 32 or 64 character by 16 line formats, and can be used with either a CRT monitor or a TV set (if you have an RF modulator) . . . **$149.95** plus $3.00 post. & insur.
  - ☐ **DeLuxe Steel Cabinet** for ASCII keyboard/terminal . . . **$19.95** plus $2.50 post. & insur.
- ☐ **New! Terminal/Monitor:** (See photo) Same features as above, except 12" monitor with keyboard and terminal is in deluxe single cabinet; kit . . . **$399.95** plus $7 post. & insur.
- ☐ **Hazeltine terminals:** Our prices too low to quote — **CALL US**
- ☐ **Lear-Sigler terminals/printers:** Our prices too low to quote: **CALL US**
- ☐ **Hex Keypad/Display kit** . . . **$69.95** plus $2 post. & insur.
- ☐ **AP-1 Power Supply Kit** ±8V @ 5 amps) in deluxe steel cabinet . . . **$39.95** plus $2 post. & insur.
- ☐ **Gold Plated S-100 Bus Connectors** . . . **$4.85** each, postpaid.
- ☐ **RF Modulator kit** (allows you to use your TV set as a monitor) . . . $8.95 **postpaid.**
- ☐ **16k RAM kit** (S-100 board expands to 64k) . . . **$199.95** plus $2 post. & insur.
- ☐ **32k RAM kit** . . . **$299.95** plus $2 post. & insur.
- ☐ **48k RAM kit** . . . **$399.95** plus $2 post. & insur.
- ☐ **64k RAM kit** . . . **$499.95** plus $2 post. & insur.
- ☐ **16k RAM Expansion kit** (to expand any of the above in 16k blocks up to 64k) . . . **$99.95** plus $2 post. & insur. each.
- ☐ **Intel 8085 cpu Users' Manual** . . . **$7.50** postpaid.
- ☐ **12" Video Monitor** (10MHz bandwidth) . . . **$139.95** plus $5 post. & insur.
- ☐ **Beginner's Pak** (see above) **$169.95** plus $4 post. & insur.
- ☐ **Experimenter's Pak** (see above) . . . **$219.95** plus $6 post. & insur.
- ☐ **Special Microsoft BASIC Pak Without Terminal** (see above) . . . **$329.95** plus $7 post. & insur.
- ☐ Same as above, plus **ASCII Keyboard Terminal With Cabinet, Get Free RF Modulator** (see above) . . . **$499.95** plus $10 post. & insur.
- ☐ **Special 8" Disk Edition Explorer/85** (see above) . . . **$1499.95** plus $26 post. & insur.
  - ☐ **Wired & Tested** . . . **$1799.95** plus $26 post. & insur.
- ☐ **Extra 8" CDC Floppy Drives** . . . **$499.95** plus $12 post. & insur.
  - ☐ **Cabinet & Power Supply For Drive** . . . **$69.95** plus $3 post. & insur.
- ☐ **Drive Cable Set-up For Two Drives** . . . **$25** plus $1.50 post. & insur.
- ☐ **Disk Controller Board With I/O Ports** . . . **$199.95** plus $2 post. & insur.
- ☐ **Special: Complete Business Software Pak** (see above) . . . **$699.96** postpaid.
  - SOLD SEPARATELY:
  - ☐ **CP/M 1.4** . . . **$100** postpaid.
  - ☐ **CP/M 2.0** . . . **$150** postpaid.
  - ☐ **Microsoft BASIC** . . . **$325** postpaid.
- ☐ **Intel 8085 cpu User Manual** . . . $7.50 **postpaid.**
- ☐ **Level "A" Monitor Source Listing** . . . **$25** postpaid.

Continental U.S.A. Credit Card Buyers Outside Connecticut

CALL TOLL FREE: 800-243-7428

To Order From Connecticut Or For Technical Assistance, call (203) 354-9375

Total Enclosed (Conn res. add sales tax) $ ________
Paid By:
☐ Personal Check ☐ Cashier's Check/Money Order
☐ VISA ☐ **Master Charge** (Bank No. _ _ _ _)
Acct. No. ________________ Exp. Date ______
Signature ________________
Print Name ________________
Address ________________
City ________________
State ________ Zip ________

NETRONICS Research & Development Ltd.
333 Litchfield Road, New Milford, CT 06776

CIRCLE INQUIRY NO. 81

some problems. For example, we will show you some trouble we had.

KEYBOARD CHARACTER? We press 'clear'

—

Oops! Screen clears, except for this. We press 'enter'.

?FC ERROR IN 130
READY
>_

Well, that didn't work! Why not? Because, the reason the 'clear' key is there is to let you clear the screen right now, whenever you press it. You cannot enter it as a keyboard character in response to an 'input' statement.

More trouble!

KEYBOARD CHARACTER? We pressed ← several times
. . .nothing happened.

If you press ← then press 'enter' you will get an 'fc error'. Recall please, what ← does. It is used to delete an incorrect character. It is a control character; it is not an everyday keyboard character. Here is another.

KEYBOARD CHARACTER? We press ↓

—

The cursor moved to here

If you now type a letter or a number or most any other keyboard character, you will get the Ascii code of that character —not the Ascii code of ↓.

Please be patient. We will tell you what we know about this stuff as quickly as we figure out what is going on. In the meantime, experiment! Here is still another.

KEYBOARD CHARACTER? __ We pressed →

Pressing → caused the cursor to move to the right. That is exactly what the control character → is supposed to do.

Wander about the keyboard. Find out about Ascii codes for keyboard characters. Save the following table of Ascii codes from 0 to 95. We will use it (and assume that you have it) in future articles in which we will suggest experiments to help us learn about what is going on in the TRS-80.

0-7	None
8	Backspaces and erases
9	None
10-13	Carriage returns
14	Turns cursor on
15	Turns cursor off
16-22	None
23	Converts to 32 character mode
24	Backspace cursor (←)
25	Advance cursor (→)
26	Downward line feed (↓)
27	Upward line feed (↑)
28	Home cursor to upper left
29	Move cursor to beginning of line
30	Erase to end of line
31	Clear to end of frame
32	Space
33	!
34	"
35	#
36	$
37	%
38	&
39	'
40	(
41	)
42	*
43	+
44	,
45	-
46	.
47	/
48-57	0 through 9
58	:
59	;
60	<
61	=
62	>
63	?
64	@

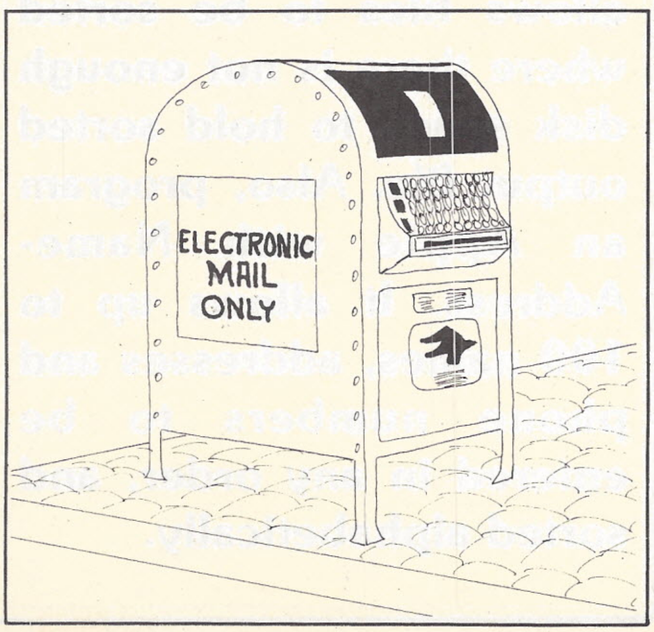
```
65-90   A through Z
   91   ↑ or [
   92   ↓
   93   ←
   94   →
   95   _ (underline)
```

Guess my letter

It's play time now. How about a game of Guess My Letter? In this game, the computer will "think" of a letter from A to Z. You try to guess what letter the computer is "thinking" of.

```
100 REM**GUESS MY LETTER GAME
110 CLEAR 200
120 A$ = "YOU GUESSED MY LETTER!!!"
130 B$ = "TRY CLOSER TO THE BEGINNING OF THE
        ALPHABET"
140 C$ = "TRY CLOSER TO THE END OF THE ALPHABET"
200 REM**EXPLAIN GAME TO PLAYER
210 CLS
220 PRINT "I'M THINKING OF A LETTER FROM A TO Z"
230 PRINT "GUESS MY LETTER!!!"
300 REM**X = ASCII CODE FOR SECRET LETTER
310 X = RND(26) + 64
400 REM**GET GUESS (G$). IS IT A LETTER?
410 PRINT : INPUT "YOUR GUESS (A TO Z)" ; G$
420 IF ASC(G$)<65 PRINT "GUESS A LETTER!" : GOTO 410
430 IF ASC(G$)>90 PRINT "GUESS A LETTER!" : GOTO 410
500 REM**COMPARE GUESS WITH SECRET LETTER
510 IF ASC(G$) = X THEN PRINT A$ : GOTO 610
520 IF ASC(G$)<X THEN PRINT C$ : GOTO 410
530 IF ASC(G$)>X THEN PRINT B$ : GOTO 410
600 REM**WINNER. ASK FOR REPLAY
610 PRINT A$
620 PRINT : PRINT "TO PLAY AGAIN, PRESS SPACE BAR"
630 KEY$ = INKEY$ : IF KEY$ = "" THEN 620 ELSE 640
640 IF KEY$ = " " THEN 210 ELSE 630
999 END
```


This is a continuing series for parents and teachers who wish to help kids learn how to use, program and enjoy the Radio Shack TRS-80 computer. The series began in the Aug-Sep 1979 issue of IA. Parts 1 through 3 are available free as an 8-page booklet from Sharon Ross, Radio Shack Circulation Dept. 3, 1300 One Tandy Center, Forth Worth, TX 76102. □



COMING NEXT MONTH

Avoid the high cost of alterations and find out how to receive more disk storage for the TRS-80 model I with a modification system that allows the use of CP/M.

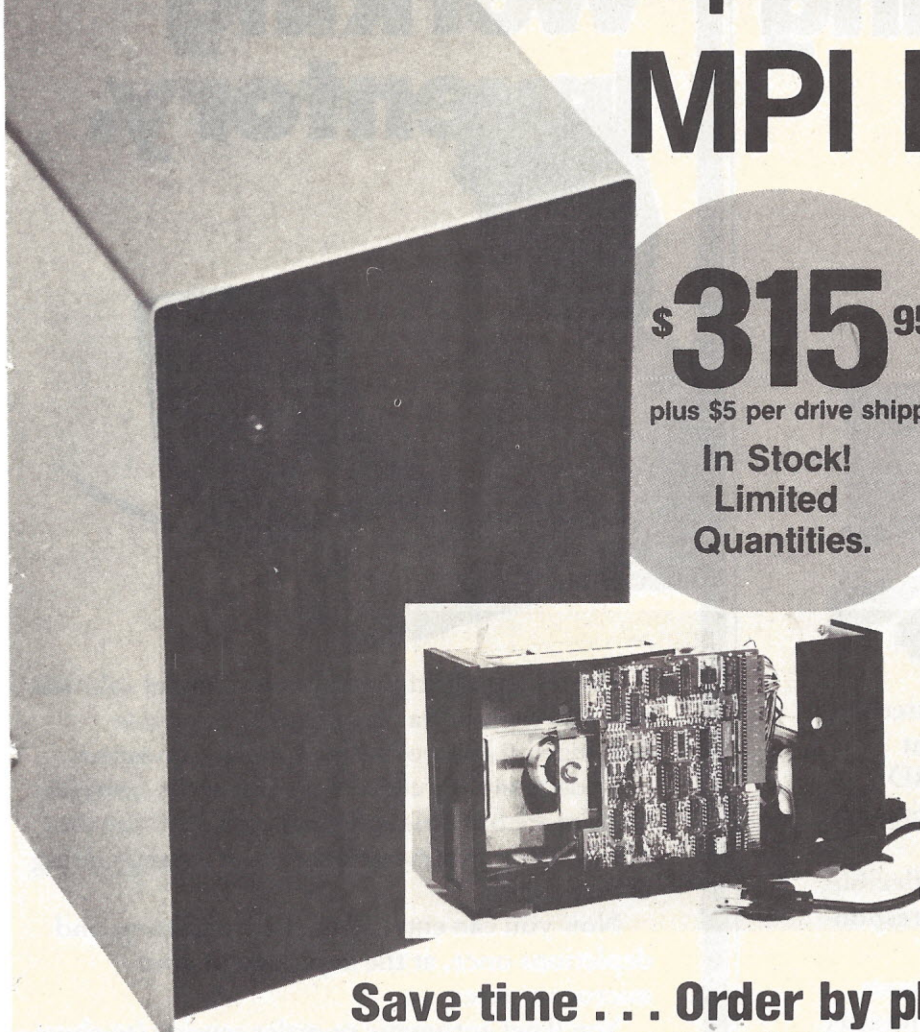
Get simple step-by-step guidance on how to use Pascal. For those already familiar with the language, some practical tips ordinarily found in complex documents and manuals are given.

Program an Apple to sort a random access disk file. Learn a technique that allows files to be sorted where there is not enough disk space to hold sorted output file. Also, program an Apple with Name-Address. It allows up to 100 names, addresses and phone numbers to be entered in any order, and sorted alphabetically.

Fully TRS-80® Compatible

MPI B/51 DISK DRIVE

FREE CATALOG Hundreds of other computer products at major savings. Cut out this coupon and send with your name and address.



$315.95
plus $5 per drive shipping

In Stock! Limited Quantities.

Includes: Case and Power Supply

Fully tested

Guaranteed for 90 days!

Here's why the MPI/B-51 is the drive for success:

- 40 tracks
- 5 ms track-to-track
- Auto-eject
- Hi-Temp stability
- Fully-closable door
- Speed constant <1½%
- Double density head
- Optical sensors—no switches
- 102K per disk

ADDS MORE POWER TO YOUR SYSTEM

Save time . . . Order by phone

Toll free: 1-800-323-4335

IN ILLINOIS CALL: 312-251-5955

TRS-80™ TANDY CORP.

Other Money Savings Opportunities Order by Phone or Mail

Cables

2 drive	$34.95
4 drive	44.95

Diskettes

Verbatim	10 for $29.95
Memorex	10 for $26.95
Georgia Magnetics	10 for $28.95
Dysan	5 for $24.95
Plastic file box	$3.95 each

Operating Systems

TRS DOS 2.2	$14.95
Percom Patch #4	$9.95
New DOS 40 track	$49.95
New DOS+ 40 track	$99.95
TRS DOS manual	$5.95

TRS-80

16K level II with keypad	$749.95
Expansion interface 0K	$269.95

Printers

Centronics 779-2	$1,050.00
Centronics 730	$899.00
IDS-440	$949.00
NEW! TX-80	$945.00
NEC Spinwriter 5530	$2,679.00

16K Memory Kits

300ns	$69.95
250ns	$74.95
200ns	$84.95
MPI Service Manual	$3.00
MPI Engineering Manual	$30.00

MIDWEST COMPUTER PERIPHERALS
P.O. BOX 437 • WILMETTE, ILLINOIS 60091

CIRCLE INQUIRY NO. 41

Midwest Computer Peripherals

P.O. Box 437 • Wilmette, Illinois 60091

master charge VISA ACCEPTED!

085

Quantity	Description	$ each	Total
		6% Ill. Tax	
		Shipping ($2.50 min.)	
		TOTAL	

(Min. Order $10.00)

☐ Check enclosed

Bill my ☐ Visa
☐ Master Charge

Acc. No. ______ Exp. ______

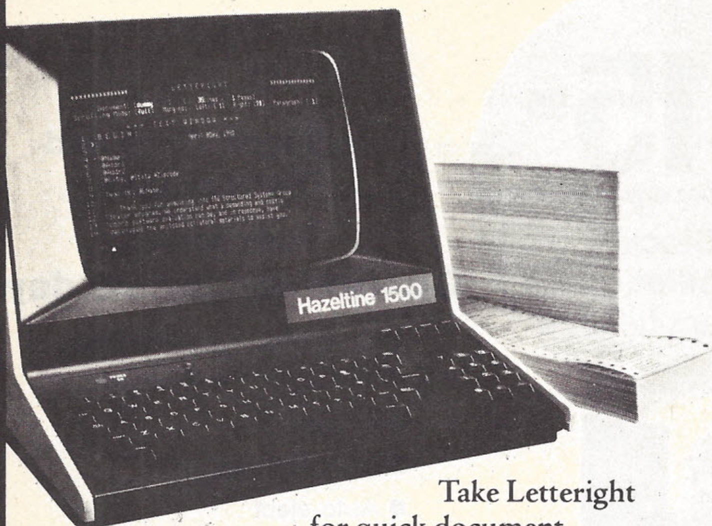
☐ Please send catalog

Name ______

Address ______

City ______ State ______ Zip ______

SSG Writing and Mailing Systems.



Take Letteright for quick document preparation and edit plus NAD Name And Address for extensive mailing list capabilities.

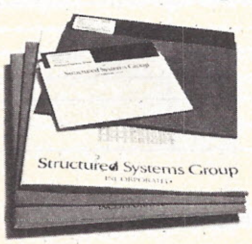
Put them together and you've got a flexible, powerful solution to big and small correspondence problems.

With Letteright you create and edit your document right on the screen. It's much easier to use than a typewriter. The letters are always perfect, and revisions are a snap.

Letteright's "wild card" slots let you create standard letters and forms, then insert information selected from your mailing list to address and "personalize" the letter.

The NAD system will store lots of names and addresses, with identifying information you create. You then print lists, labels, or envelopes of virtually any group you want from the list, or the whole list.

This pair should be working for every microcomputer owner.



Letteright and NAD are part of a full line of working software solutions from Structured Systems Group, all ready to run on any CP/M® microcomputer system. CP/M is a registered trademark of Digital Research.

Structured Systems

5204 Claremont Oakland, Ca 94618 (415) 547-1567

CIRCLE INQUIRY NO. 95

The Working Inventory.

Here's a straightforward stock control solution that will replace stacks of paperwork while helping you control your expensive inventory.

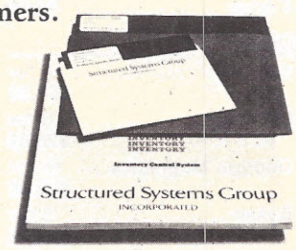
Today's money costs make inventory carrying costs more burdensome than ever. And the cost of inventory control personnel isn't exactly going down, either.

Now you can enter your stock additions and depletions once, at the keyboard of your microcomputer.

You'll get automatic re-order reports (to show you what you need), complete inventory parts listings and valuations (to show you what you have), and activity reports (to show what's moving).

You'll worry less, because you'll know more. The Inventory Control System includes an auditability option, extensive error screening, and reliable documentation. It was designed for people, not for programmers.

Call or write SSG for a complete System Summary and the names of your closest dealers. There are over 150 nationwide.



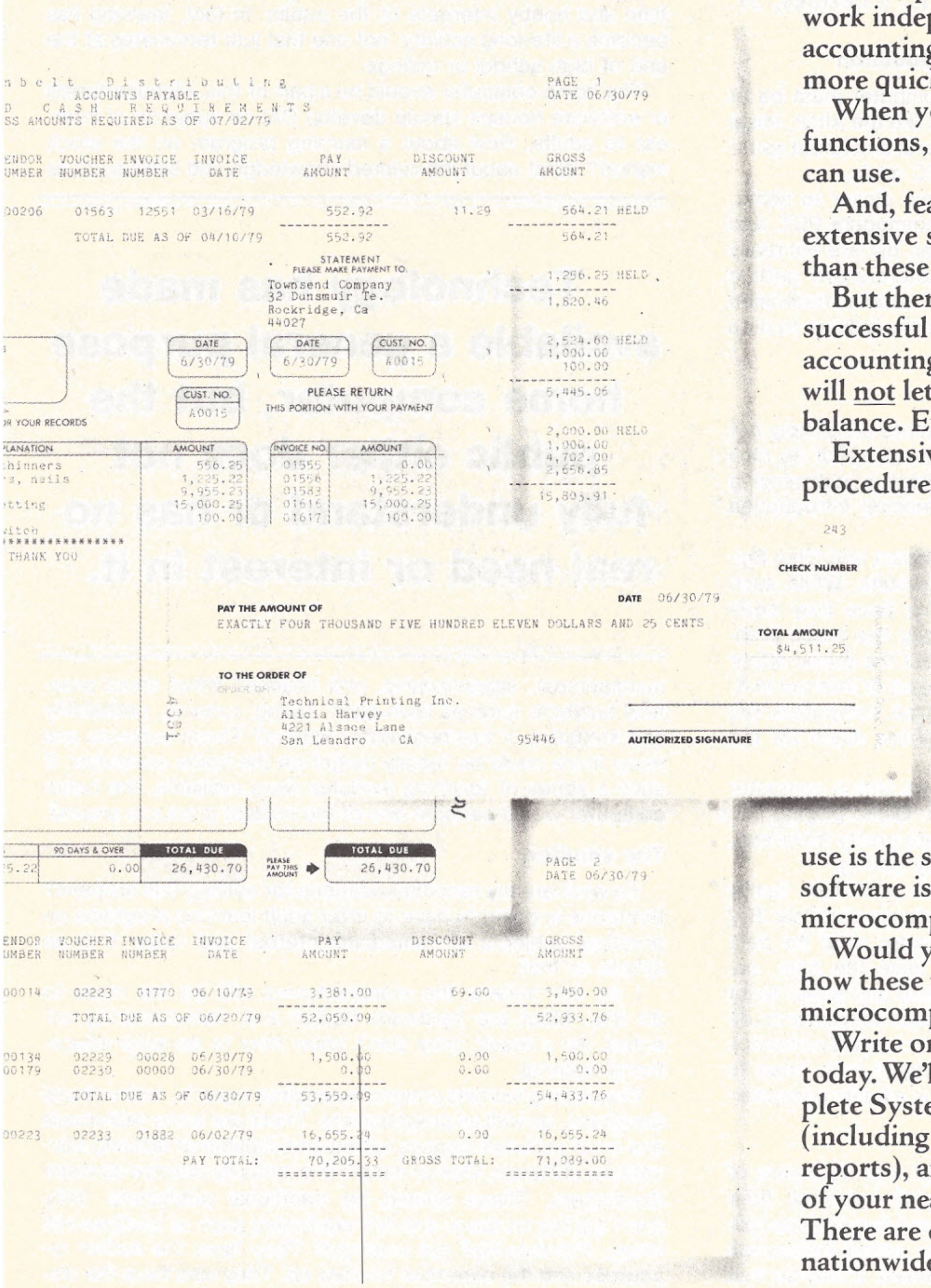
The Inventory Control System is part of a full line of working software solutions from Structured Systems Group, all ready to run on any CP/M® microcomputer system. CP/M is a registered trademark of Digital Research.

Structured Systems

5204 Claremont Oakland, Ca 94618 (415) 547-1567

CIRCLE INQUIRY NO. 96

The Working General Ledger, Accounts Receivable, Accounts Payable.



Three proven, refined software packages that work independently or together to simplify your accounting procedures, produce statements more quickly, and track information better.

When you computerize these important functions, you naturally look for features you can use.

And, feature for feature, you won't find more extensive solutions running on microcomputers than these three systems.

But there's a single feature essential to the successful business system. Reliability. SSG accounting is reliable. For example, the systems will not let your operator or files get out of balance. Ever.

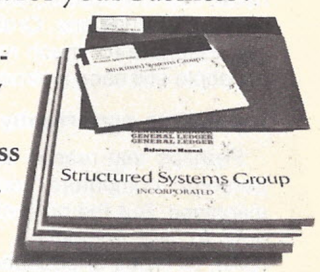
Extensive error-checking and file back-up procedures built into the software reduce errors, and protect your valuable time and data.

Enter your transaction once, and file it. Thorough audit trails ensure data integrity from source document to financial statement, as the computer quickly and accurately finishes your bookkeeping job.

This reliability is guaranteed, because the software system you use is the same as the master copy at SSG. Your software is maintained and updated by a leading microcomputer software vendor.

Would you like more details and specifics on how these three software systems can put a microcomputer to work for your business?

Write or call SSG today. We'll send a complete System Summary (including sample reports), and the address of your nearest dealers. There are over 150 nationwide.



G.L., A/R, and A/P are all part of a full line of working software solutions from Structured Systems Group, all ready to run on any CP/M® microcomputer system. CP/M is a registered trademark of Digital Research.

CIRCLE INQUIRY NO. 53

Structured Systems Group INCORPORATED

5204 Claremont Oakland, California 94618 (415) 547-1567

Learning with Micros

By Louis E. Frenzel, Jr.

The Home Computer and Education

The idea of a true home consumer computer must be at least five years old. The original machines like the Altair, Imsai and the SWTP 6800 were hobby computers, and about as far from the home computer as the early DEC minis.

The second generation computers came closer to home. These were the Radio Shack TRS-80, Commodore PET, and the Apple II. These are enormously popular, but are not really consumer computers. However, manufacturers are getting closer. The Atari 400 and 800, TI's 99/4, Mattel's Intellivision and the new APF are approaching what many believe to be a real home computer.

Where are the buyers?

But while the hardware is here, consumers are simply not buying. The TI 99/4 is not selling well because of its high price, despite discounting and rebates. Atari is just ramping up production, and Mattel is still experiencing introduction delays of its keyboard unit.

Numerous tests have been made by Interact and Ohio Scientific in department stores with mixed results. While such tests have not been total failures, neither have they been truly successful or conclusive. Technology has made available a general purpose home computer, but the public either does not fully understand or has no real need or interest in it.

Why? Price is a major factor. Most home computers still sell in the $600 to $1,500 price range—too much for the average consumer.

It may also be that the things one can do with a computer are things many people don't care to do. Game playing is a main application. But most of the games are pretty trivial and, after the novelty wears off, can get pretty boring.

Keeping your checkbook, Christmas card list, family budget and calendar on a computer is certainly practical. But it takes time to accumulate, record and enter all of the data required. The average person doesn't have the time, patience or perseverence when it is a lot faster and easier to jot notes in a checkbook, and use a $10 pocket calculator to do banking calculations. Ordinary 3x5 cards or a spiral notebook are adequate for much recordkeeping. What it boils down to is people still need a stronger reason to buy a home computer.

A possible opportunity

Perhaps that reason is education. Most manufacturers of personal computers tout the educational value of their machines. But the potential goes far beyond teaching people how to use computers. Using computer aided instruction (CAI), all types of education can be presented. It may be that education is the one practial application that will entice consumers to buy a home system.

Most home computer manufacturers offer educational software. But most are extremely short and trivial programs for children: basic arithmetic, spelling, subjects that most kids learn in school with little difficulty. They certainly don't need a home computer. Some programs I have seen are even too simple for children. Kids don't want to come home from school and take a learning program in math or spelling. I think the home computer manufacturers are on the wrong track with their educational software. Forget the kid stuff and get on to developing something that adults can use.

There is a mini revolution going on in education these days. Adults are going back to school in droves. They are seeking all kinds of educational materials, sources and experiences. Colleges are opening up more evening programs and continuing education classes.

Most of this renewed interest is job or career oriented. People want to do their jobs more competently, prepare for advancement, or change careers. Doctors, lawyers, pharmacists and engineers work hard at continuing education to keep up with rapidly changing technology and society.

Adult education appeals to the general interest, leisure time and hobby interests of the public. In fact, learning has become a life-long activity, not one that just terminates at the end of high school or college.

The home computer should be a part of this. Manufacturers or software houses should develop CAI on subjects of interest to adults. How about a learning program on the stock market? What about job-related knowledge and skills such as management, salesmanship, and finance? What about practical subjects such as lawn care, sailing, running, probability and statistics or business letter writing? These subjects and many more could be readily taught on the home computer. If such a series of teaching modules were available, the home computer would become one of the hottest products around.

Technology has made available a general purpose home computer, but the public either does not fully understand or has no real need or interest in it.

The solution

So what are my recommendations for solving this problem? Someone is going to have to offer adult learning programs on practical subjects of immediate interest and value. It is as simple as that.

I do not believe the manufacturers will be the ones to do this. Most are hardware design and manufacturing oriented. As a result, they don't know how to develop educational material.

Developing learning programs requires experts, instructional designers, as well as programmers. There are some individuals and small software houses willing and capable of tackling educational software. One or more will eventually emerge as major developers. These should be traditional publishers. Why don't the big textbook and A/V publishers such as McGraw-Hill Wiley, Prentice-Hall get involved? They have the author resources and the expertise to draw on. They also have the distribution systems to make such programs widely available. No doubt some kind of relationship between a manufacturer and publisher could result in some extremely interesting adult learning programs for home computers. This type of education is what the home computer needs to be successful. □

Pascal/MT® +

The Choice of Professionals, Worldwide

Pascal/MT + is an enhanced version of our successful Pascal/MT package popular with both hardware and software engineers. Generating even more optimized ROMable 8080/Z80 object code (directly from the Pascal source), Pascal/MT + now includes those features demanded by today's professional. We've added modular compilation and linking (incorporating the industry standard linker format used by FORTRAN and PL/I) and user selectable Z80 native code or our enhanced 8080/Z80 object code. Also included is an improved SYMBOLIC DEBUG, more I/O facilities and even an option to produce pseudo assembly output intermixed with the Pascal source.

MODULAR COMPILATION

Pascal/MT + generates the same industry standard relocatable code used in FORTRAN and PL/I. Pascal code may be linked with our linker or any compatible linker. Both Pascal and assembly language modules may be separately compiled and then combined to produce a final program. With modular compilation available, the run time overhead becomes as small as 256 bytes, and is typically 1200 bytes.

OPTIMIZED NATIVE OBJECT CODE

Pascal/MT + now produces even more optimized object code than before. Now included is a compile-time selectable option to generate Z80 code where it is more efficient. The 8080/8085 code is better, too.

IMPROVED DEBUGGING FACILITIES

The Pascal/MT + debugger is improved. We've added facilities to use the debugger with larger programs and even in a ROM-based environment. The compiler can also output a disassembled listing of your programs with the Pascal source code interspersed between the object code. This helps you gain visibility in non-CP/M® applications.

ENHANCED I/O CAPABILITIES

To our already powerful console, printer and re-directable I/O facilities we have added Pascal standard TEXT file I/O as well as improved line editing for console I/O.

PLUS . . .

- Transcendental functions
- Improved String Handling
- Benchmark tests rate our code up to 10 times faster than P-code Pascal and faster than other native code high level languages too!
- Re-assembly of the run-time package no longer required for ROM based applications!

AND OUR STANDARD FEATURES:

- High speed compilation (up to 2000 lines per minute)
- Non-CP/M® environments supported
- CP/M® file support
- Program chaining
- Built-in mini-assembler
- Compile-time constants
- Business (18-digit) arithmetic
- Scientific (6.5 digit) arithmetic
- AMD9511 Hardware support
- Source code for run-time package included (re-assembly now requires Macro-80 assembler)
- I/O port, Interrupt facilities
- Bit and Byte manipulation
- Many useful built-in procedures (loaded from run-time library)

SYSTEM REQUIREMENTS

To execute the Pascal/MT + system you must have a CP/M® system with at least 32K of memory. For larger applications we recommend 48K or more. Source programs are prepared using ED (or equivalent), we recommend WORD-MASTER.

(Coming soon: versions for Intel ISIS-II and Heath HDOS. Ask us.)

ORDERING

STILL ONLY $250
will buy you one of the best software power tools available.

Pascal/MT + has been chosen by companies such as GE, FMG and Chromatics (as well as over 700 other companies) as the Pascal for them, why not join them!

The Pascal/MT + package:

- BCD compiler
- Floating point compiler
- Linker
- Interactive Symbolic Debugger
- Run time package in Source and Object form.
- Pascal library utility routines
- 95 + page user's guide
- Sample programs

The user's guide is available for $30.00, refundable with a system purchase.

MT microSYSTEMS has a very reasonable graduated, one time royalty arrangement for free standing software generated by Pascal/MT. Ask us.

To order Pascal/MT + write or call:

MT MicroSYSTEMS

1562 Kings Cross Drive
Cardiff, CA 92007

(714) 753-4856

We ship on 8" single density and 5-1/4" North Star single density. Other formats available. Ask us.

THE MICRO-MATHEMATICIAN

by Dr. Alfred Adler

Rotation of Coordinate Axes

Mechanics is that branch of physics which concerns itself, among other things, with the motion of matter. This matter is in the form of objects or bodies which can either be elastic or rigid. A rigid body is one in which the special relationships between the particles making up the body are fixed with respect to one another.

In the three dimensional world, rigid bodies can move in six different ways, each independent of the other. That is to say, they have six degrees of freedom. Thus a rigid body can move fore and aft, sideways, and up and down. These motions, called translations, are independent of each other since they occur at right angles or perpendicular to each other. There are also three rotations, referred to in aircraft and boats, as roll, pitch, and yaw. These motions similarly are perpendicular to each other and therefore mutually independent.

The three translations, since they are mutually perpendicular, occur along three mutually perpendicular lines. Customarily these lines are defined coincident with, or perpendicular to, lines or planes of symmetry of the body. Such axes are referred to as body axes; that is they are fixed in the body and move with it. This choice presents a number of advantages in discussing the geometry of the motion. The rotational motions are also mutually perpendicular, and the axes of rotation are customarily chosen coincident with these same lines for the same reasons.

Uncoupling the motions

If these same lines are chosen as the initial coordinate axes, the mathematical description of the geometry is simplified, and a number of complex terms in the equations of motion drop out. These are terms which relate or couple the rotational motions to each other. Finally, if these coordinates pass through the center of mass of the body, the rotational motions are decoupled from the translational motions, and we end up with six independent degrees of freedom.

Positions of the forces and moments which produce the translational or rotational motions shed some light on the choice of axes. Referring to figure 1, we see in (a) that a translational force that does not pass through the center of mass produces rotation, as opposed to (b), in which a translational force that passes through the center of mass produces a pure translational motion.

In figure 2(a) a torque is imposed whose axis does not pass through the center of mass. Such a torque produces not only rotation, but also translation. In (b) we have located the axis of the torque through the center of mass. This eliminates the translational motion; however, due to the coupling between the rotational motions mentioned previously, the axis of the rotation will not coincide with the axis of the torque. If it is desired that the rotation take place about the same axis as the imposed torque, we must let the axis of the torque not only pass through the center of mass, but also coincide with an axis or plane of symmetry of the body, as in (c).

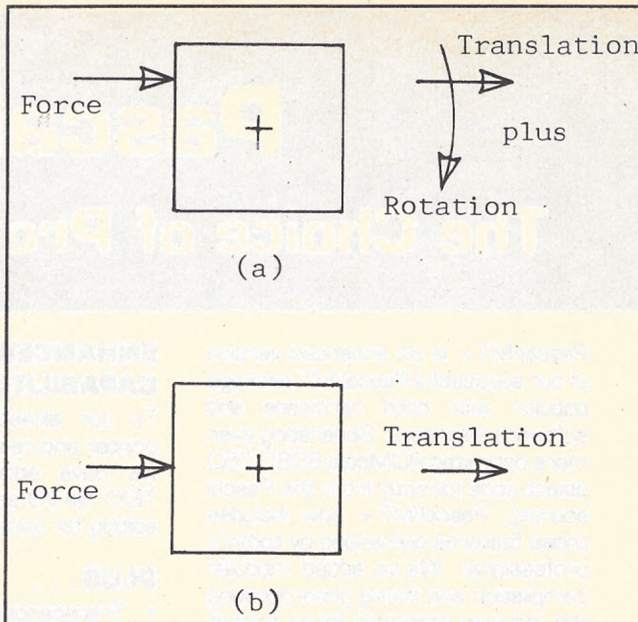


Figure 1. Translational force (a) not passing through center of mass, (b) passing through center of mass.

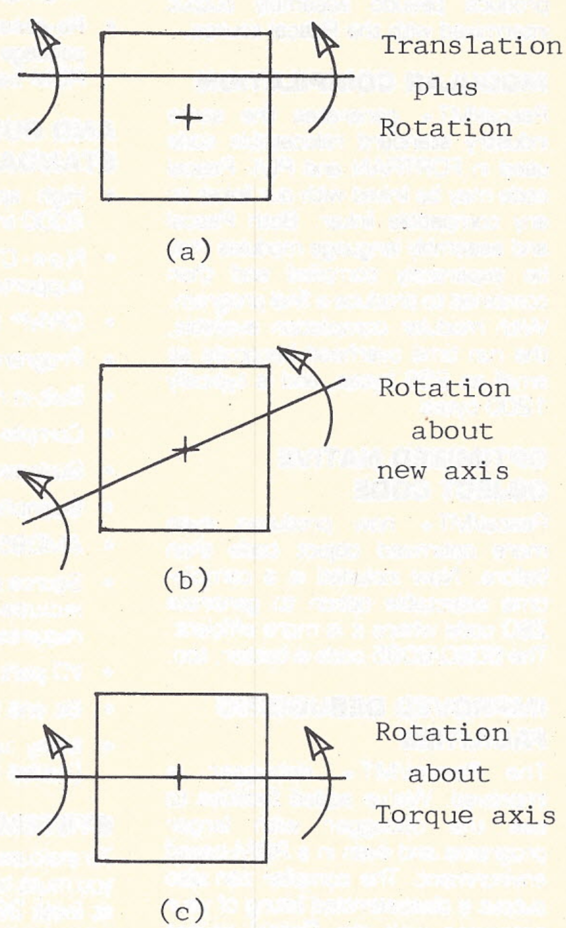


Figure 2. Torque (a) not passing through center of mass, (b) passing through center of mass but not aligned with principle axis, (c) passing through center of mass and aligned with principal axis.

What happens in the physical world is not dependent on man's choice of coordinates, rather it is dependent on the manner in which the forces and torques are imposed. Thus whether or not the forces and torque axes pass through the center of mass definitely affects the motion. The choice of coordinate axes on the other hand affects man's representation of the motion; that is, the equations of motion and the coupling of the motions. The choice of coordinate axes permits the equations of motion to reflect independence of the six degrees of freedom and leads to the simplest possible description of the motion.

Inertial axes

So far, the best possible choices have been made. Unfortunately, there are several other factors that have not yet been addressed. Once motion occurs, if we continue to use the body axes, we will be dealing with a set of coordinates which themselves are being accelerated. Newton's laws of motion hold only in inertial, or nonaccelerated coordinates. Either fictitious translational acceleration terms and fictitious rotational terms must be added to correct for the acceleration of the coordinate system, or else an inertial coordinate system must be used. The latter course is often preferable.

The inertial coordinate system may be initially coincident with the body axis system. If gravity is important, it might be established coincident with the vertical and horizontal. In the analysis of aircraft motion, the X axis is in the wind direction, the Z axis is vertical, these are called 'wind' axes. In any event, either the motion is analyzed directly in the inertial system, or it is analyzed in the body axis system (which is easier) and then correction terms are added.

Ultimately, there is no escape from these terms; it is simply a matter of where, when and how you want to deal with them. If the latter course is taken, it is necessary to be able to define the location of a point on the body in inertial coordinates. Since the orientation of the body in body axes is fixed, this requires only that the location of the body axes be defined in the intertial system. Only three angles are required and it is desirable that these three angles be independent of each other. The customarily used Euler's angles are such a set.

Surprising results

As an indication that the problem is not trivial, the following experiment should be tried: Take a book laying flat on the table and imagine the x axis in the plane of the book, running from top to bottom, the y axis running from side to side, and the z axis running vertically from the center of the book. Rotate the book 90° about the x axis and follow this with a 90° rotation about the y axis. Remember, these are body axes and therefore rotate with the book. The book is now in a vertical plane with the x axis running left to right, the y axis vertically, and the z axis away from the reader.

Repeat the experiment, but this time do the rotation about the y axis first and follow this with the rotation about the x axis. This time the books ends up in a vertical plane with the z axis running from left to right, the x axis vertically, and the y axis away from the reader. As a quick insight into the many ramifications of this unexpected development, let it simply be stated that finite angular displacements are not vectors.

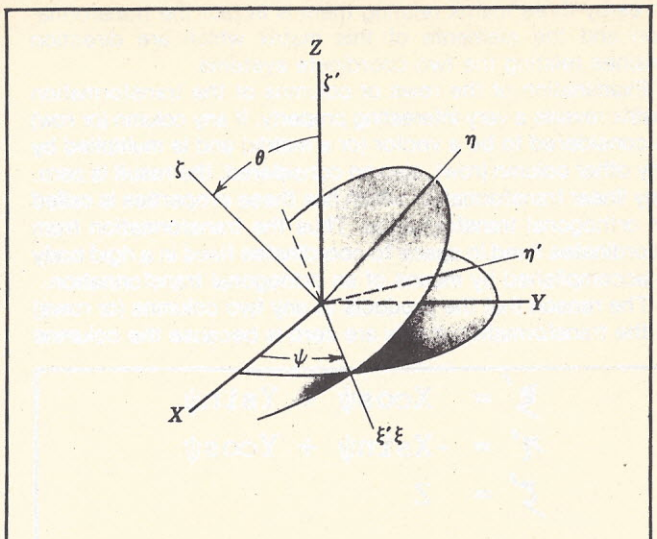


Figure 4. Rotation about Z axis through angle θ.

Euler's angles are three independent angles capable of defining the position of the x, y, z body axes relative to the X, Y, Z inertial axes. Start with the two sets of axes coincident. All rotations are counterclockwise when viewed from the positive end of the pertinent axis of rotation.

Referring to figure 3, the body axes are first rotated through an angle Ψ about the Z axis so as to take up new positions ξ', η', and ζ'. This rotation is commonly referred to as yaw. Equations relating these new positions to the X, Y, Z set are given in figure 6. The second rotation, shown in figure 4, is by an angle ϑ about the new ξ' axis, such that the new positions of the body axes are ξ, η, and ζ. This rotation is referred to as pitch. Equations relating the ξ, η, and ζ positions to the ξ', η', and ζ' positions are also given in figure 6. Finally, in figure 5, the body axes are rotated by an angle σ about the ζ axis, taking up their final positions x, y, and z. This rotation is called roll. The equations describing this rotation are also given in figure 6.

Orthogonal transformations

The three sets of equations presented in figure 6 are rewritten in matrix form in figure 7. Each of the three matrix equations is a transformation equation. The column matrix on the right is in fact a vector, with scalar elements X, Y, Z in the case of the first equation. The column matrix on the left is similarly a vector, with scalar elements ξ', η', and ζ'. The

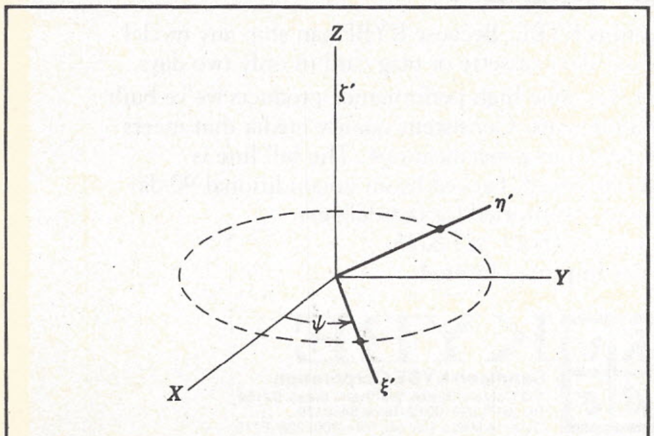


Figure 3. Rotation about Z axis through angle Ψ.

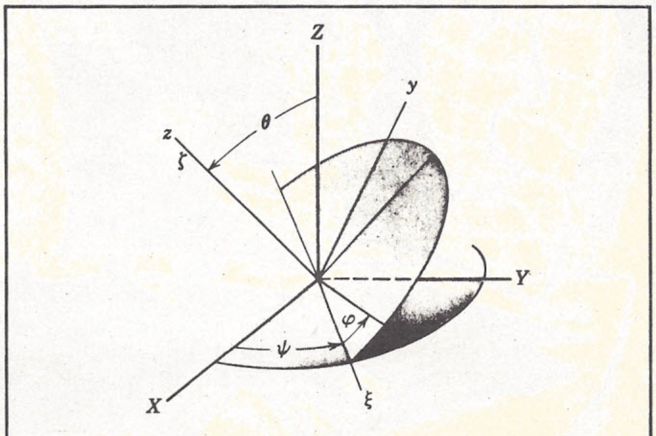


Figure 5. Rotation about Z axis through angle φ.

three-by-three matrix relating them is in fact the transformation and the elements of this matrix which are direction cosines relating the two coordinate systems.

Examination of the rows or columns of the transformation matrix reveals a very interesting property. If any column (or row) is considered to be a vector (or a matrix) and is multiplied by any other column (row), also so considered, the result is zero. Any linear transformation which has these properties is called an orthogonal transformation. Thus the transformation from coordinates fixed in space to coordinates fixed in a rigid body is accomplished by means of an orthogonal transformation.

The reason that the products of any two columns (or rows) in the transformation matrix are zero is because the columns

$$\xi' = X\cos\psi + Y\sin\psi$$
$$\eta' = -X\sin\psi + Y\cos\psi$$
$$\zeta' = Z$$

$$\xi = \xi'$$
$$\eta = \eta'\cos\theta + \zeta'\sin\theta$$
$$\zeta = -\eta'\sin\theta + \zeta'\cos\theta$$

$$x = \xi\cos\varphi + \eta\sin\varphi$$
$$y = -\xi\sin\varphi + \eta\cos\varphi$$
$$z = \zeta$$

Figure 6. Equations describing rotations of figures 3, 4, and 5.

$$\begin{bmatrix} \xi' \\ \eta' \\ \zeta' \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$\begin{bmatrix} \xi \\ \eta \\ \zeta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \xi' \\ \eta' \\ \zeta' \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \xi \\ \eta \\ \zeta \end{bmatrix}$$

Figure 7. Equations of figure 6 in matrix form.

(or rows) considered as vectors are orthogonal (at right angles) to each other. Also, the way they are multiplied, considered as matrices, is equivalent to taking the dot (or scalar) product of them as vectors; and the dot product of orthogonal vectors is zero. Thus orthogonality of the transformation means that the coordinate systems are orthogonal; that is, they consist of three mutually perpendicular vectors, or axes, and they remain so under the transformation (the rotation).

Since the product of two orthogonal transformations is itself an orthogonal transformation, we can multiply the three transformation matrices in figure 7 into a single transformation matrix which in one stroke transforms the inertial X, Y, Z

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} (\cos\phi\cos\psi - \sin\phi\cos\theta\sin\psi) & (\cos\phi\sin\psi + \sin\phi\cos\theta\cos\psi) & (\sin\phi\sin\theta) \\ (-\sin\phi\cos\psi - \cos\phi\cos\theta\sin\psi) & (-\sin\phi\sin\psi + \cos\phi\cos\theta\cos\psi) & (\cos\phi\sin\theta) \\ (\sin\theta\sin\psi) & (-\sin\theta\cos\psi) & (\cos\theta) \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Figure 8. Complete transformation from X, Y, Z to x, y, z.

coordinate system into the body axis x, y, z system. This transformation equation is given in figure 8.

If the transformation matrix of figure 8 is inverted, we obtain the inverse transformation from body axes to the inertial system. This transformation is shown in figure 9.

Program Eulerang

Since the transformations given in figures 7, 8 and 9 are tedious to work with at best and present maximum opportunity for error, a program has been written to do the dog work.

In the program, Eulerang, the notation had to be changed since most microcomputers do not support the Greek alpha-

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} (\cos\varphi\cos\psi - \sin\varphi\cos\theta\sin\psi) & (-\sin\varphi\cos\psi - \sin\psi\cos\theta\cos\varphi) & (\sin\theta\sin\psi) \\ (\cos\varphi\sin\psi + \sin\varphi\cos\theta\cos\psi) & (-\sin\varphi\sin\psi + \cos\varphi\cos\theta\cos\psi) & (-\sin\theta\cos\psi) \\ (\sin\theta\sin\varphi) & (\sin\theta\cos\varphi) & (\cos\theta) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Figure 9. Complete transformation from x, y, z to X, Y, Z.

bet. The new notation is completely covered in REMs at the beginning of the listing. The user is given the option of converting from inertial to body axes, or from body to inertial axes. If the former is chosen, the further option is presented of printing the results of the intermediate rotations, angle by angle, or going directly to the complete transformation. Whichever options are chosen, the current coordinates and the three transformation angles must be loaded. The program then responds with the new transformed coordinates.

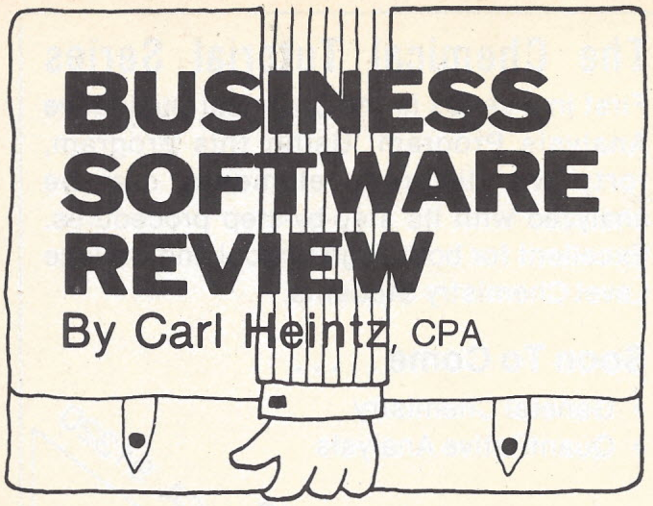
The sample runs have been made using a different number for each variable to maximize the likelihood of catching the type of errors most likely to be made by anyone keyboarding this program. An inverse transformation is included also as a check.□

PROGRAM LISTING

```
5 REM*********************************************************
10 REM
15 REM*********  P r o g r a m   E U L E R A N G  *********
20 REM
25 REM**********  Version  1.0  ****  June 1980  **********
30 REM
35 REM*******  Written by - Alfred A. Adler, Ph.D. ********
40 REM
45 REM  *    *    *    *    *    *    *    *    *    *    *    *    *
50 REM Inertial or "Wind" axes are X,Y,Z
55 REM xi',eta',zeta' are axes after rotation psi about Z
60 REM xi,eta,zeta are axes after rotation theta about xi'
65 REM x,y,z are axes after rotation phi about zeta
70 REM  *    *    *    IN THIS PROGRAM    *    *    *    *    *
75 REM psi,theta,phi will be denoted by A1,A2,A3
80 REM X,Y,Z will be denoted by X(1),X(2),X(3)
85 REM xi',eta',zeta' will be denoted by X1(1),X1(2),X1(3)
90 REM xi,eta,zeta will be denoted by X2(1),X2(2),X2(3)
95 REM x,y,z will be denoted by X3(1),X3(2),X3(3)
100 REM *    *    *    *    *    *    *    *    *    *    *    *    *
105 REM
110 DIM A1(3,3),A2(3,3),A3(3,3),B(3,3)
115 R=57.295780
120 !"Do you want to transform (A) from X,Y,Z to x,y,z; or"
125 INPUT"   (B) from x,y,z to X,Y,Z ? : ",B$
130 IF B$="B" THEN 480
135 REM
```

```
140 REM *** FORWARD TRANSFORMATION - X,Y,Z to x,y,z ***
145 REM
150 INPUT"State X,Y,Z coordinates : ",X(1),X(2),X(3)
155 !"Euler's angles are measured counterclockwise, in"
160 !"   degrees, about the Z, xi', and zeta axes."
165 INPUT"State psi,theta, and phi : ",D1,D2,D3
170 A1=D1/R\A2=D2/R\A3=D3/R
175 !"In addition to the X3, the X1 and X2 can be printed."
180 INPUT"Do you want to see the X1 and X2 ? :",A$
185 REM
190 REM *** SET UP TRANSFORM MATRIX FOR X TO X1 ***
195 REM
200 A1(1,1)=COS(A1)\A1(1,2)=SIN(A1)\A1(1,3)=0
205 A1(2,1)=-SIN(A1)\A1(2,2)=COS(A1)\A1(2,3)=0
210 A1(3,1)=0\A1(3,2)=0\A1(3,3)=1
215 REM
220 REM *** COMPUTE THE X1 ***
225 REM
230 FOR K=1 TO 3\X1(K)=0\NEXT
235 FOR I=1 TO 3
240 FOR J=1 TO 3
245 X1(I)=X1(I)+A1(I,J)*X(J)
250 NEXT J
255 NEXT I
260 REM
265 REM *** SET UP TRANSFORM MATRIX FOR X1 TO X2 ***
270 REM
275 A2(3,2)=-SIN(A2)\A2(3,3)=COS(A2)\A2(3,1)=0
280 A2(2,2)=COS(A2)\A2(2,3)=SIN(A2)\A2(2,1)=0
285 A2(1,2)=0\A2(1,3)=0\A2(1,1)=1
290 REM
295 REM *** COMPUTE THE X2 ***
300 REM
305 FOR K=1 TO 3\X2(K)=0\NEXT
310 FOR I=1 TO 3
315 FOR J=1 TO 3
320 X2(I)=X2(I)+A2(I,J)*X1(J)
325 NEXT J
330 NEXT I
335 REM
340 REM *** SET UP TRANSFORM MATRIX FOR X2 TO X3 ***
345 REM
350 A3(1,1)=COS(A3)\A3(1,2)=SIN(A3)\A3(1,3)=0
355 A3(2,1)=-SIN(A3)\A3(2,2)=COS(A3)\A3(2,3)=0
360 A3(3,1)=0\A3(3,2)=0\A3(3,3)=1
365 REM
370 REM *** COMPUTE THE X3 ***
375 REM
380 FOR K=1 TO 3\X3(K)=0\NEXT
385 FOR I=1 TO 3
390 FOR J=1 TO 3
395 X3(I)=X3(I)+A3(I,J)*X2(J)
400 NEXT J
405 NEXT I
410 REM
415 REM *** PRINT ***
420 REM
425 !
430 !"X,Y,Z = ",TAB(20),%10F4,X(1),X(2),X(3)
435 IF A$(1,1)<>"Y" THEN 450
440 !"xi',eta',zeta' = ",TAB(20),%10F4,X1(1),X1(2),X1(3)
445 !"xi,eta,zeta  = ",TAB(20),%10F4,X2(1),X2(2),X2(3)
450 !"x,y,z = ",TAB(20),%10F4,X3(1),X3(2),X3(3)
455 !
460 GOTO 120
465 REM
470 REM *** INVERSE TRANSFORM - x,y,z to X,Y,Z ***
475 REM
480 INPUT"State x,y,z coordinates : ",X3(1),X3(2),X3(3)
485 !"Euler's angles are measured counterclockwise, in"
490 !"   degrees, about the X, xi', and zeta axes."
495 INPUT"State psi,theta, and phi : ",D1,D2,D3
500 A1=D1/R\A2=D2/R\A3=D3/R
505 REM
510 REM *** SET UP ROW 1 OF INVERSE TRANSFORM MATRIX ***
515 REM
520 B(1,1)=COS(A3)*COS(A1)-SIN(A3)*COS(A2)*SIN(A1)
525 B(1,2)=-SIN(A3)*COS(A1)-SIN(A1)*COS(A2)*COS(A3)
530 B(1,3)=SIN(A2)*SIN(A1)
535 REM
540REM *** SET UP ROW 2 OF INVERSE TRANSFORM MATRIX ***
545 REM
550 B(2,1)=COS(A3)*SIN(A1)+SIN(A3)*COS(A2)*COS(A1)
555 B(2,2)=-SIN(A3)*SIN(A1)+COS(A1)*COS(A2)*COS(A3)
560 B(2,3)=-SIN(A2)*COS(A1)
565 REM
570REM *** SET UP ROW 3 OF INVERSE TRANSFORM MATRIX ***
575 REM
580 B(3,1)=SIN(A2)*SIN(A3)
585 B(3,2)=SIN(A2)*COS(A3)
590 B(3,3)=COS(A2)
595 REM
600 REM *** COMPUTE THE X ***
605 REM
610 FOR K=1 TO 3\X(K)=0\NEXT
615 FOR I=1 TO 3
620 FOR J=1 TO 3
625 X(I)=X(I)+B(I,J)*X3(J)
630 NEXT J
635 NEXT I
640 REM
645 REM *** PRINT ***
650 REM
655 !
660 !"x,y,z = ",TAB(20),%10F4,X3(1),X3(2),X3(3)
665 !"X,Y,Z = ",TAB(20),%10F4,X(1),X(2),X(3)
670 !
675 GOTO 120
READY
```

```
RUN

Do you want to transform (A) from X,Y,Z to x,y,z; or
   (B) from x,y,z to X,Y,Z ? : A
State X,Y,Z coordinates : 1,2,3
Euler's angles are measured counterclockwise, in
   degrees, about the Z, xi', and zeta axes.
State psi,theta, and phi : 30,40,50
In addition to the X3, the X1 and X2 can be printed.
Do you want to see the X1 and X2 ? :Y

X,Y,Z =                 1.0000    2.0000    3.0000
xi',eta',zeta' =        1.8660    1.2321    3.0000
xi,eta,zeta  =          1.8660    2.8722    1.5062
x,y,z =                 3.3997     .4167    1.5062

Do you want to transform (A) from X,Y,Z to x,y,z; or
   (B) from x,y,z to X,Y,Z ? : B
State x,y,z coordinates : 3.3997,.4167,1.5062
Euler's angles are measured counterclockwise, in
   degrees, about the X, xi', and zeta axes.
State psi,theta, and phi : 30,40,50

x,y,z =                 3.3997     .4167    1.5062
X,Y,Z =                 1.0000    2.0000    3.0000

Do you want to transform (A) from X,Y,Z to x,y,z; or
   (B) from x,y,z to X,Y,Z ? : B
State x,y,z coordinates : 1,2,3
Euler's angles are measured counterclockwise, in
   degrees, about the X, xi', and zeta axes.
State psi,theta, and phi : 30,40,50

x,y,z =                 1.0000    2.0000    3.0000
X,Y,Z =                 -.5918    -.7536    3.6169

Do you want to transform (A) from X,Y,Z to x,y,z; or
   (B) from x,y,z to X,Y,Z ? : A
State X,Y,Z coordinates : -.5918,-.7536,3.6169
Euler's angles are measured counterclockwise, in
   degrees, about the Z, xi', and zeta axes.
State psi,theta, and phi : 30,40,50
In addition to the X3, the X1 and X2 can be printed.
Do you want to see the X1 and X2 ? :N

X,Y,Z =                 -.5918    -.7536    3.6169
x,y,z =                 1.0000    2.0000    3.0000

Do you want to transform (A) from X,Y,Z to x,y,z; or
STOP IN LINE 125
READY
```



A Peach of a General Ledger Program

The Peachtree general ledger program by Retail Sciences, Atlanta, GA, is part of a series of accounting packages including accounts receivable, accounts payable, payroll and inventory. The programs are written in Microsoft Basic, which means that the user is given the source code. This allows for customization of specific user changes if the user is familiar with programming.

The first thing the purchaser will be impressed with is the professional manual. Nicely organized in a 3-ring binder with tabs, the manual is easy to get into. There are three sections: a brief description of the system, an operator's manual, and a set of special aids designed to assist in implementation. And, of course, there are numerous appendices with explanations to error messages and data file descriptions.

The bulk of the manual is devoted to a detailed description of the programs, how they work, their options and alternatives. The manual literally leads the user through screen after screen, with detailed explanations behind each step. Even though the programs themselves are largely self-explanatory due to the numerous menus, the manual rounds out each step. No prior experience with microcomputer programs is necessary for a user to be successful on a first pass.

System requirements

The program is designed for any 8080, 8085 or Z80 computer with 48K RAM and two disks each with at least 250K. The video terminal must have 80 characters per line with 24 lines and scrolling capabilities. The system printer must have at least 132 column print capabilities. The system runs under Microsoft disk Basic under CP/M.

If a 48K system is used, up to 400 accounts can be on the system. The number of transactions per month is limited only by the physical disk space. For example, with a chart of 200 accounts, there is room on a single density 8-inch diskette for about 1,200 transactions.

A systems walkthrough

The first time a user logs on, the system must be initialized. This program sets system parameters, sets up account codes (ranges) and creates new data files (empty files). It can do any one or all three. System parameters include the clear screen codes, number for the system and data drives, and whether the user wishes to have automatic transfer from other Peachtree programs. The system allows automatic input from the payroll, accounts payable and accounts receivable programs. Only the Peachtree programs, or custom programs producing the same kinds of files, may be used.

Account number sequences are up to five digits. There are no limits as to what numbers are assets, liabilities, or whatever; the user specifies.

Initialization is also used to set up journal types. There can be up to nine journal types, including repeating journal entries made each month, prior period journal entries, cash receipts, cash disbursements and general journal. The user can specify the names for the journal codes.

Master that file

Entering the chart of accounts is done with a program entitled Master File Maintenance that allows the user to add, change, delete, or query an account. The system allows the formatting of financial statements at the same time accounts are entered. Some accounts are balance accounts, while the remainder are formats for the financial statements. This is similar to programs sold on some minicomputers, and not dissimilar to the Structured Systems Group general ledger. Peachtree's system is extremely flexible, allowing for columnar formatting, totaling, subtotals and titles to be completely user defined. The financial statements look extremely professional.

The master file also contains fields for the current amount year-to-date and budget amounts, and has a space for the balances for each month in the previous year. Thus, comparative financials are possible.

After the accounts have been entered, they may be queried or deleted. (Deletions are not possible for accounts with balances.) The query program operates rapidly and is quite easy to use. Accounts may be listed in one of three options:

1. List all the information for each account, including the dollar amounts for current, year-to-date, budget and the previous 12 fiscal months.
2. As above, except without the previous 12 months or;
3. Only the chart of accounts.

Since the list program is separate, it can be run without going through all the masterfile set-ups.

Data input

The speed and convenience of data input are critical to the success of any general ledger program, and a good benchmark of the program's overall suitability. This journal entry sequence is among the best available. It is quick, easy and flows naturally. The conventional debit and credit conventions are used; debits are positive numbers, credits are entered with a '–' preceding. Amounts may be up to $9,999,999.99. Dollar signs of columns may be entered, if that is easier on the user. The entry sequence has a nice default option allowing for the repeating of a field. After each journal entry, the user is asked about any changes, so errors can be trapped and changed before they are entered. At the end of an entry session, a control report lists what has just been entered.

Once a transaction is started, it cannot be cancelled. The user can just enter carriage returns until the sequence is finished, then delete transactions using a special program. While this is not as convenient as the ability to delete an erroneous entry when it has just been made, this is compensated by the ability to modify entries as they are put in.

Transaction listing

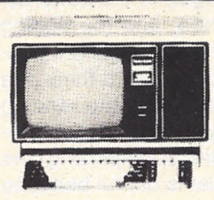
Once all the transactions have been entered, a transaction list can be produced containing all data entered since the last posting. The list can be organized sequentially by account number, or by source code (separate journals by source are easily produced). This feature allows an accountant to have the journals just like a manual system, and makes for very easy editing. Alternately, the sequential journal list is useful. Since it lists all data by account number, an accountant familiar with what entries should be present can easily make changes before financials are produced.

Often a user needs to query the status of only one or two accounts. The Query Account Status program allows a user to view one account at a time with the beginning balance, all

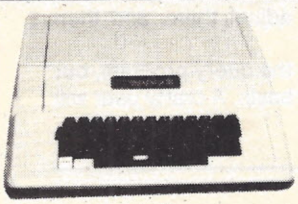
ΩMEGA SALES CO.

"WHOLESALE COMPUTER PRICES" DIRECT TO THE PUBLIC

12 Meeting St., Cumberland, R.I. 02864



TRS•80
Model II - $3,500



Apple II
16K - $1049

INTERTEC SUPERBRAIN
32K RAM - $2449.00
64K RAM - $2649.00

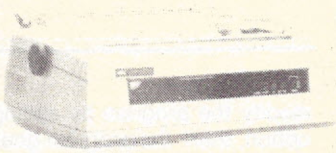
PRODUCT SPECIAL OF THE MONTH!!

Atari 800
$749.00

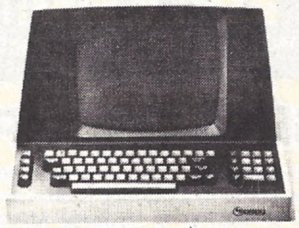
Products are NOW IN STOCK AT ΩMEGA Sales Co.

For: TRS•80, Apple, CBM
(Interface Included)

Epson TX-80 - $745



NEC Spinwriter
5510-5530 - $2449



Soroc 120 - $699

CALL TOLL FREE FOR ΩMEGA'S PRICE!

ΩMEGA OFFERS THE BEST DELIVERY AND PRICE ON:
APPLE • ATARI • TRS•80 MODEL II • INTERTEC •
T.I. 810 • HEWLETT-PACKARD-85 • SOROC •
COMMODORE • NEC • QUME • CENTRONICS

CIRCLE INQUIRY NO. 45

ΩMEGA sells only factory fresh, top quality merchandise to our customers.
ΩMEGA will try to match any current advertised price with similar purchase conditions.
Before you buy anywhere else - be sure to call ΩMEGA Sales Co.
1-401-722-1027 or

MEGA
OLL FREE

1-800-556-7587

ΩMEGA ships via UPS, truck, or air. COD's, VISA, Mastercharge accepted.

"A member in good standing of the better business bureau"

VISA master charge

transactions posted, and a current-to-date balance. The output can be to the printer or video display.

Trial balance

This program performs several functions. The program name is somewhat misleading since what the program does, basically, is update the master file. The program also prints a detail general ledger as it posts, containing all of the detail activity for the month, the beginning balance, and the ending balances. This is slightly different from the accountant's conception of what a trial balance is. Most accountants think of a trial balance as a list of each account with the ending balances shown. Some general ledger systems will also provide a beginning balance and one figure summarizing the change or activity for the month. Such a list is useful because adjusting entries and whatnot may be overlooked in the first pass. In this system, the user is given complete list with all the detail. Unfortunately, at the time of preparation, all activity detail is purged. Thus, there is no convenient way to make entries, view the resulting balances, adjust them, and then get a printout with all the detail.

The way to get around this is to use the query program, but that is done on an account-by-account basis. A crafty user will

...file errors occur due to hardware failures or operators who exit programs in irrational or random ways.

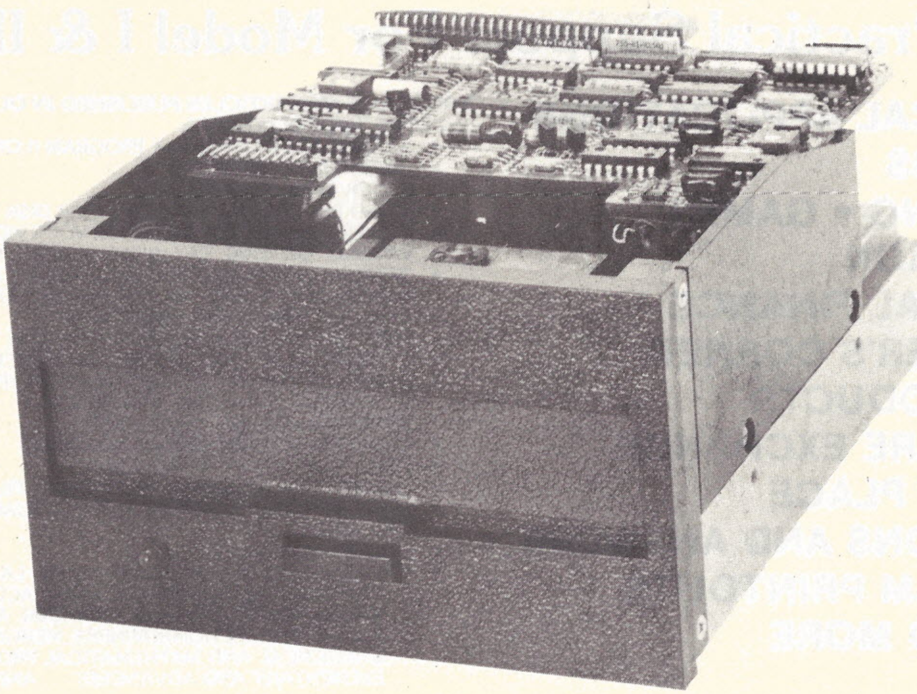
modify the program by adding four lines of simple code. On option, then, the program goes into a loop listing all accounts.

The financial statement programs are straightforward and easy to use. They contain options which allow for inclusion of prior year data or to have departmental income statements. Since the formatting of the financials is largely up to the user, many combinations are possible. However, due to the method by which the financials are created, all accounts appear in the financials sequentially, i.e., in the same order they appeared in the chart of accounts. Of course, several accounts can be combined in presentation on the financials.

A unique and useful feature is the verify file structure program which examines all data files to assess whether file links have been properly maintained. If there are problems, the program will link to a Fixit program what will rebuild the files. Generally, file errors occur due to hardware failures or operators who exit programs in irrational or random ways.

Often, users desire source code for programs with the idea of instigating major changes or customizations to the programs. In most cases, users of Peachtree's software will use it as it comes, with the possible exception of the one change mentioned. The buyer should keep in mind that these programs are long, complex and sophisticated. They utilize what retail sciences call Maris or multi-array ISAM, a sophisticated method of file management which most professionals prefer. The ISAM is implemented in Basic.□

Get On the "Band" Wagon



A Review of Micro Peripherals' Mini-Floppy Disk Drive

by Roger H. Edelson

While not exactly brand new, I have just tried one of MPI's family of 5¼-inch floppy disk drives. All its drives, ranging from 250K bytes to 1 megabyte, use a unique approach to the conversion from stepper-motor rotary output to the linear movement required by the head positioning transport. Instead of rotary lead-screw, or cam and cam-follower approaches, MPI uses a patented pulley-band technique much like that used to convert rotary tuning knob motion to linear movement of a slide-rule dial on many tuners.

MPI has improved on floppy dial-cord design by utilizing a flexible metal band with one-half slotted to let the other, thinner, half pass through. The band center is secured to the drive pulley and each end is connected to the head positioning block (figure 1). In order to avoid band slackness and the attendant backlash, both sides are pretensioned during assembly and attached to the head positioning block. While the MPI patent covers a method using either a compression or an expansion spring, it found that a fixed value of pretension produced superior results when the drives were subjected to a range of operating temperatures.

The use of a drive pulley with a firmly attached band allows the rotational movement of the stepper motor to be converted directly into linear head motion through the mechanism of the drive pulley radius. The equation expressing this relationship is:

$$d = (a/360) \times \pi \times 2R$$

where

d = linear displacement
a = rotational movement in degrees
R = drive pulley radius

Therefore, with a pulley radius of 0.3316 inch, the nominal 3.6° rotation of the stepper-motor translates to a head movement of 0.20803 inch (equivalent to 48 TPI). As can be seen by this simple equation, the only factors affecting accuracy of the head position are the

tolerance of the drive pulley radius (which can be easily held to within 0.001 inch) and accuracy of angular displacement of the stepper motor.

MPI indicates that this allows significantly improved (<5ms.) track-to-track access time by reducing both rotational and linear inertia. The specifications for the four models in the MPI line of mini-floppies are given in the table. Notice that the average track access time for even the 96 TPI models (91 and 92) is still well under the relatively standard figure of 288 msec (the actual figure is 150 msec), and that of the 48 TPI models is a fast 84 msec.

MPI mini-floppies reflect the previous experience of the company founders in designing full-sized floppies for GSI and Orbis. All four models are equipped with fully-closing front doors and disk ejector mechanisms just like their 8-inch cousins (figure 2).

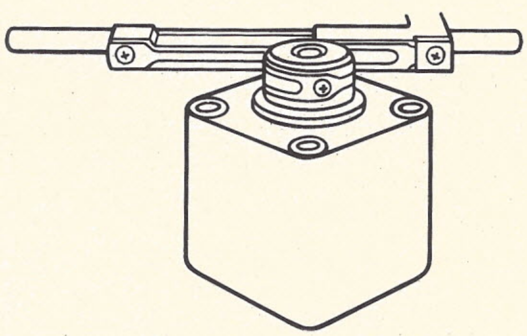


Figure 1. Flexible band and drive pulley.

A door lock option and drives with a Shugart-style front door are offered for OEM users. MPI is quite serious in its attempt to gain a share of the OEM market.

Testing the performance

How well does the drive perform? I tried the model 52, MPI's double-sided double-density 48 TPI. It uses the head-loading arrangement adopted by most manufacturers featuring a fixed lower head and a moving upper head to load the media. This is an improvement over the earlier clothespin-style head-loading mechanisms which appreciably reduced media life. With the execution of the double-head assembly, this drive is the same as the model 51 single-sided drive and uses most of the same mechanical parts and electronics. As this drive is industry/ANSI standard as far as interface cable connections, power cable connections, and mechanical size are concerned, it was very easy to substitute for one of my present drives. After about 15 minutes of work—most of which involved removing and replacing the cover of my drive assembly cabinet—I was up and running.

Insertion and removal of the diskette are smooth and easy. The disk ejector mechanism is a nice touch; besides making it very easy to remove a disk, it also reduces or eliminates disk center crunching. Unless the diskette is correctly and firmly seated, it will be forced gently back into your hand and the door cannot be closed. I like the fully closing door. Even with positive air pressure in my drive assembly, I worry about dust damage.

Another nice touch is LEDs and photosensors for track 00 and write protect sensing. Besides being very reliable, photosensing gives a solid, quiet feel to the disk insertion process. Correct disk insertion is signalled by the small but audible click of the catch on the disk ejector mechanism. With the design of this drive, the diskette is completely unloaded when the door is opened and it can be easily removed and inserted, even though the drive activity light is on. I find this feature particularly useful when copying or formatting significant numbers of disks in my North Star Horizon—I hate to wait the 10 + seconds for the drive to dropout. However, because of the peculiarities of design, as long as the drive activity light is on, the disk ejector will not function.

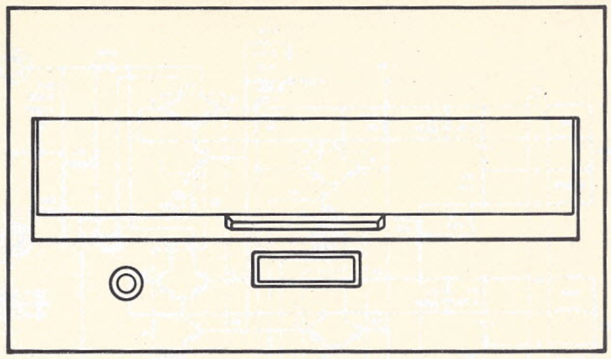


Figure 2. Door design of MPI mini-floppy.

Modifying the software

In order to use the faster track-to-track access time of the MPI drive, it is necessary to change the North Star and CP/M software by modifying the configuration byte, a not too difficult task using DDT and Saveuser for CP/M. It is nice to have this configuration byte to use for this change as it allows each drive in a 4-drive system to have different characteristics insofar as seek time and number of sides. The CP/M drive assignment of the MPI drive location was then specified as a

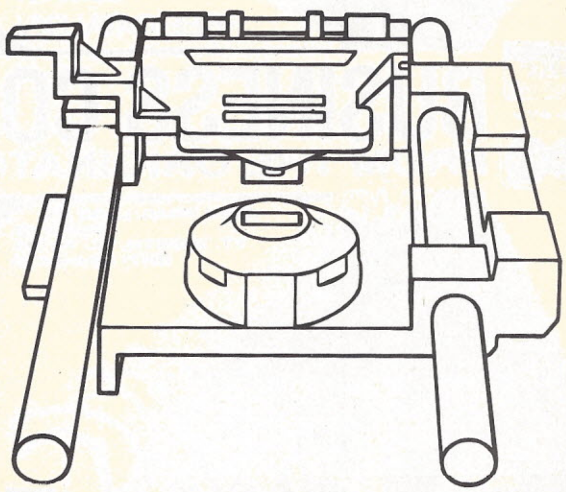


Figure 3. Head mounting arm replaced by structure between two parallel rods.

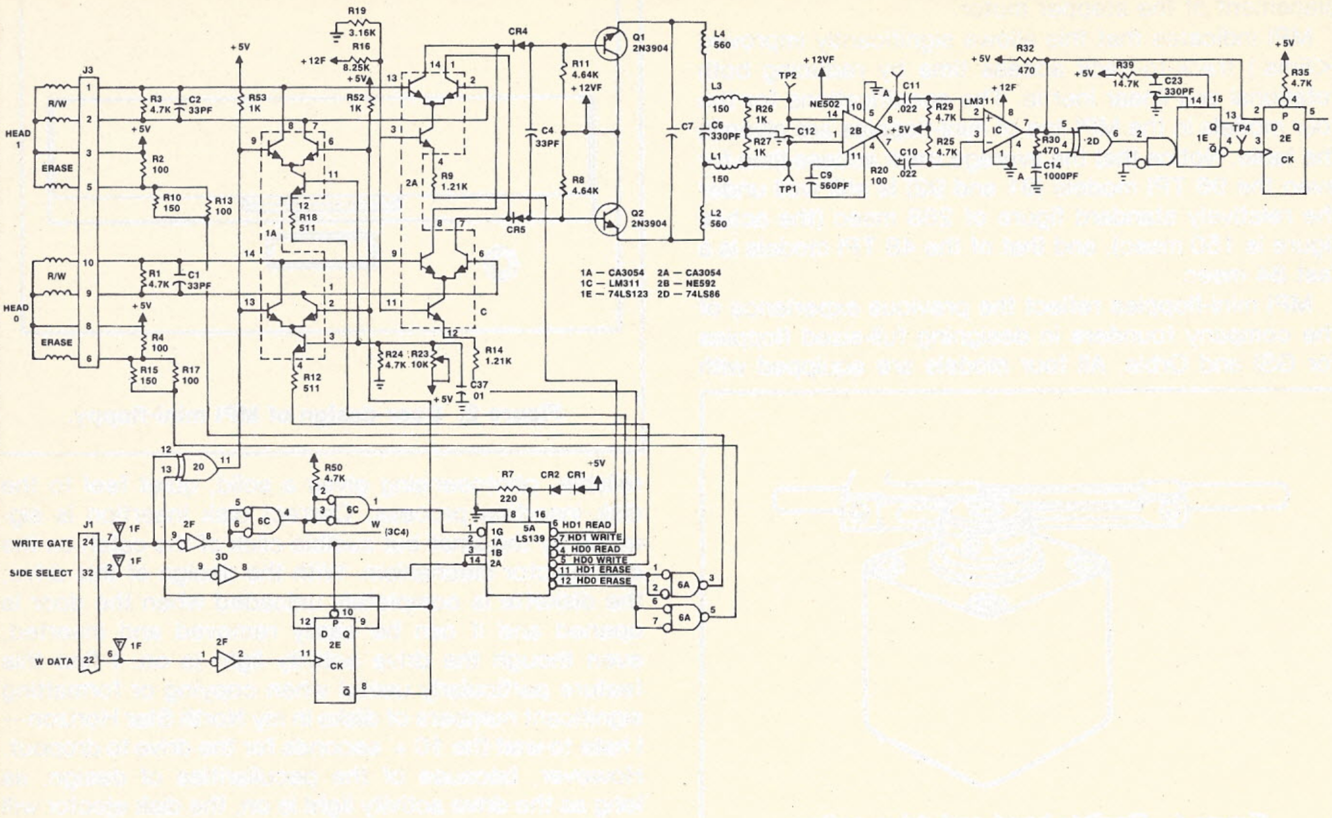


Figure 4. Read/write circuitry.

 CIRCLE INQUIRY NO. 17

high-speed-seek drive and, in subsequent use, I was able to noticeably improve my file handling benchmarks.

MPI also manufactures a 96 TPI double-sided version that stores up to 1 megabyte of unformatted data—not bad for a 5¼-inch disk. The main changes are a different stepper-motor design to allow smaller rotation for each commanded step and a different head spindle bearing. The spindle and bearing have been machined to the tighter tolerances needed for a 96 TPI drive.

The 96 TPI models use an insignificantly greater recording density than the 48 TPI drives (5922 BPI vs. 5876 BPI), so one would expect them to have the same data error rates as the more standard models. In fact, MPI quotes equal error rates for all four models—numbers equivalent to those throughout the industry. To use this drive will require major modifications to the DOS to allow access to higher track numbers. This change cannot be made in the North Star or CP/M DOS by simply changing the configuration byte.

Throughout testing and use, the drive performed flawlessly and I was particularly pleased with the door design and disk ejector mechanism. MPI claims its design incorporates a longer clutch assembly to reduce or eliminate disk centering errors. I cannot report on whether it is better than the designs of other drives, but it was not hampering and operated without failure. I suspect that the longer the clutch, the easier it is on the disk center hole when it takes up the slop in the diskette housing.

One other major design change was to replace the cantilevered head mounting arm with a structure centered between two parallel rods. They indicate that this design (figure 3) reduces head radial positioning errors and eases manufacturing problems.

Good documentation—for a price

I do take MPI somewhat to task for providing the technical manual as an extra-charge option. It is hard to believe that it costs $30 to print. I managed to beg a copy and it is quite a piece of technical literature: a complete, 106-page, product manual starting with a general description of the drive, covering installation, interface requirements, theory of operation, and ending with maintenance and troubleshooting.

The circuitry used in the MPI 51/52 series is not particularly unusual except that it does use modern, up-to-date, techniques and ICs. Head switching, both from head 1 to head 2 and from 'read' to 'write,' is done by RCA CA3054 dual independent differential amplifiers. The read/write circuitry is shown in figure 4. Control of the head switching is accomplished by an LS139 demultiplexer. With this technique, head common mode noise rejection can be around 90 dB and head-to-head isolations of >50 dB can be maintained. These figures indicate how much noise will be introduced in the disk read process. The high common mode rejection and isolation figures, along with the 8 dB noise figure of the CA3054 itself, contribute to low data error rates.

I visited the MPI factory recently and was impressed with its dedication to reliability and quality control. Drives undergo an extensive amount of testing prior to the final 8-hour run. On top of the usual quality checks, the marketing department samples each outgoing lot and repeats some quality checks. If the sample fails inspection, shipments are held up until the problem is corrected.

Currently in the works is a one-half thickness 8-inch drive that should be on the market this year. □

Specifications for four MPI models

	51	52	91	92
Capacity Specifications (In 10^3 bytes)				
Single Density (unformatted)				
Per Disk	125	250	250	500
Per Track	3.13	3.13	3.13	3.13
Double Density (unformatted)				
Per Disk	250	500	500	1000
Per Track	6.25	6.25	6.25	6.25
Functional Specifications				
Tracks	40	80	80	160
Track Density (TPI)	48	48	96	96
Recording Density (max. BPI)				
Single Density	2768	2938	2788	2961
Double Density	5536	5876	5576	5922
Rotational Speed (RPM)	300 ± 1½%			
Average Latency	100ms			
Transfer Rate				
Single Density	125K bits/sec			
Double Density	250K bits/sec			
Encoding Method	FM, MFM, M^2FM			
Access Time				
Track to Track	5ms			
Average	84ms (51/52)		150ms (91/92)	
Head Settling Time	15ms			
Head Load Time	35ms			
Power-Up Delay	0.5sec			
Interfacing	Industry/ANSI compatible			
Physical Specifications				
Environmental				
Operating Temperature	40°F to 115°F (4.4°C to 46.1°C)			
Non-Operating Temperature	-40°F to 160°F (-40°C to 71°C)			
Operating Humidity	20% to 80% (non-condensing)			
Non-Operating Humidity	5% to 95% (non-condensing)			
Electrical				
DC Power	+12V ± 5%, 0.8 Amp (1.5 Amp surge) +5V ± 5%, 0.4 Amp			
Power Dissipation	12W operation, 6W standby			
Mechanical				
Height	3.25 in. (82.6 mm)			
Width	5.75 in. (146.1 mm)			
Length	7.75 in. (196.9 mm)			
Weight	3.0 lbs. (1.36kg) for 51/52 3.1 lbs. (1.41kg) for 91/92			
Media Requirements				
Diskette	Industry compatible, 5¼ in.			
Sectoring	Soft; 10, 16			
Type	Single or Double Sided Single or Double Density			
Reliability Specifications				
MTBF	9,200 hours			
MTTR	0.5 hours			
Design Life	5 years			
Media Life	3×10^6 passes/track			
Data Integrity				
Soft Errors	1 per 10^9 bits read			
Hard Errors	1 per 10^{12} bits read			
Seek Errors	1 per 10^6 seeks			

UL and CSA Approved

Options
Door Lock
Shugart Front Door (Add -S)
Detailed design data available on request

Polymorphic Twinsystem 88

by Tom Fox

Poly's back! We were afraid it was lost. . .along with such pioneers as Imsai and Processor Technology. Born in microcomputing's dark ages—1975—Polymorphic Systems of Santa Barbara, CA was one of the architects of the small computer industry. It was one of the first to jump onto the S-100 bus, and bravely shouldered itself near the top of the heap of hobby computer manufacturers. Just as its products were making the transition from home to office applications, newcomers began offering functionally identical equipment at prices this small company couldn't match. A decision to resist the CP/M tide sealed its doom.

But it refused to give up. After filing for bankruptcy, Polymorphic emerged with a financially healthy glow, building and testing computers and shipping them by the truckload.

Oldtimers remember the Poly 88—a slim, bright orange box containing a 5-slot S-100 motherboard and enough circuit cards to make up a serviceable stand-alone computer. Recently, a 10-slot chassis was added, fitting into a solid walnut enclosure. This housing is the basis for all current systems including the single-disk 8810, 3-disk system 8813 and Twinsystem 88. The systems share more than external appearance: The internal electronics and software are so similar that a program written for one will run on any other (if the data storage space is available).

Hardware with several options

The system consists of the walnut-encased housing and a pair of CRT terminals. (For deluxe installations, an impressive walnut-grained desk is available.)

The system is floppy-disk based with a somewhat confusing choice of disk storage options. The basic computer enclosure holds from one to three 5¼-inch minifloppy diskette drives, either Shugart single-sided or Tandon double-sided. All are double density, giving a capacity of 89,600 or 179,200 bytes (characters) per diskette. Total capacity of on-line disk storage is over half a megabyte.

To add more capacity, plug in a system 88/MS, a similar-sized box containing two full-size (8-inch) floppy disk drives by Shugart. These can be specified as either single- or double-sided, and are double-density units. Storage capacities are either 625K kilobytes or 1.25 megabytes per diskette. A total of two system 88/MSs can be connected, yielding up to 5 megabytes of data capacity.

Well-vented chassis

Inside the processor box is the 10-slot S-100 motherboard placed directly behind the 5¼-inch diskette drives. All but two of the slots are filled. The left side of the enclosure is taken up by a DC power supply and cooling fans. Two low-velocity fans send cool air through slots on the right side of the chassis, past the vertically-mounted circuit cards and over the power supply; it exhausts out the left side. The linear power supply features varistor circuitry for protection from damaging spikes on the AC power line. The motherboard bus lines are terminated to solve the electrical ringing problem that showed up on earlier S-100s.

The processor box features a key-locking main power switch and 'load' button. A lamp can be wired to flash on when the processor is in use, or when it is idle and waiting for an interrupt.

The CPU circuit card utilizes an 8080A microprocessor chip running a 1.87 MHz and featuring maximum usage of processor interrupt circuitry. This is a particular advantage since it keeps the two users out of each other's way. The CPU card contains a bootstrap program in ROM to reset the system after power-up or when a programming error causes the machine to crash.

A pair of 48K RAM cards are included—one for each user. They utilize low-power dynamic memory chips and sport a parity circuit for error detection—an

unusual feature in this class of equipment. A third memory card is incorporated to run the timesharing programs. It is actually a 48K card fitted with only ⅙ the number of memory chips. The memory allocation is as follows: The first 8K is occupied by the bootstrap ROM, display refresh and floppy disk controller buffers; the next 48K (from 8K through 56K) is the address space shared by the two users (each with a separate physical memory card); and the top 8K by the timesharing software residing in its own RAM card.

Separate single-card controllers are provided for the 5¼- and 8-inch floppy diskette drives. Up to four drives can be connected to each controller. If the system is fitted with a mix of the two sizes, it is possible for one of the two users to be operating with data on a 5¼-inch diskette at the same time the other is manipulating files on an 8-inch diskette. In this case, the two controllers operate independently, giving each user unimpeded access to his own files. If both users are working with one size of diskette, they must compete for the diskette controller; and, in the worst case, contend for the physical positioning of the read/write head on the diskette drive.

Two more slots in the S-100 motherboard are taken up with a pair of identical video display controllers. Most of the terminal intelligence is in this controller card, as the terminal itself is little more than a bare shell with CRT monitor and keyboard circuitry. The screen holds 16 lines of 64 characters each; just half the number of characters featured by most 24 x 80 displays. Perhaps this is the reason the screen display seems so bright, well-focused and rock-steady. The display's powerful graphics capability, unfortunately, is barely exploited by the software.

The CRT's most impressive aspect is that it is memory mapped, rather than serial. Polymorphic is one of several manufacturers who have chosen this design path, which carries several important advantages in spite of its greater complexity.

In-house software development

Polymorphic has always chosen to do as much software development as possible in-house. The fundamental operating system, simply called 88, is single-user. It is compatible with the special needs of the memory mapped terminal and supports a single serial hardware port for driving an external printer. The operating system occupies 5 of the 48K, relying heavily upon an overlay scheme to pull routines in from a diskette as needed. The system is "user friendly" in several ways:

1. Characters can be typed ahead on the keyboard without losing them.
2. Fundamental keyboard editing functions work the same at all times—even within Basic and the various applications programs.
3. It is possible to create command files—lists of keyboard commands that execute automatically in sequence as a result of entering a single word.
4. File names can be upper and lower case, and long, descriptive names can be assigned.
5. The file directory structure contains a branching feature familiar to Unix users: Any file can be a directory to other files, permitting a great number on one diskette. Such directories can be nested almost without limit.
6. Files deleted by mistake can be undeleted with a single command.
7. Files can be selectively protected from alteration or deletion and there's even a way to encrypt certain types of data files.

The two users execute identical copies of the operating system, with control switched back and forth 60 times per second. One user can force programs to run in the other user's time slot and memory space—very handy even for those all alone with the machine.

Highly efficient text editor

This software ranks with the best. It is screen oriented, meaning the text is visible on the screen for manipulation by judicious use of the keyboard. All changes made to the text are instantly visible. This is perhaps the most impressive showpiece for the memory mapped display. Most of today's screen-oriented text editors go to great pains to simulate a memory

Serial vs. Memory-Mapped Terminals

Terminals characterized as serial receive characters via a single wire from the CPU. Whole displays must be squeezed through this funnel a single character at a time, although the rate can be very fast—960 characters per second or more. Once a character is passed to the terminal, its internal circuitry flashes it onto the same position on the screen 60 times each second, lest it fade away from view.

In memory mapped terminals, this function is brought closer to the CPU. Screen updates are executed between refresh cycles, and any number of characters can be added, deleted or changed at essentially the same time. This results in an effective data transfer rate of tens of thousands of characters per second, although it is not strictly proper to speak in such terms when referring to memory mapped displays.

The term memory mapped comes from that portion of the main memory address space dedicated to the display function alone. Programs that need to display characters simply write them into the appropriate memory address, and the video display card includes the new character in the next screen update. This requires that each program know the screen location of each character before it is sent there—a burden not shared by software for serial terminals. For Polomorphic machines, this is taken care of with special software routines, so most users don't have to worry about it.

mapped display within the inherent limitations of a serial terminal with varying degrees of success.

One of the newer products is Wordmaster II, a set of programs that ties the text editor into a page formatter and menu-driven command processor for office applications. Wordmaster II can perform nearly all the functions of the operating system in a simple, generously-prompted manner by novice operators. For example, the file directory can be altered as if it were another text file, and the files are renamed, deleted or copied instantly to match the altered directory.

Basic is the major language on the system. This one was described to us as "your standard enhanced superset of Dartmouth Basic." In a world dismally lacking in Basic standards, we find a lot that is similar among the various offerings. This version is neither more nor less capable than most. However, it holds the booby prize in execution time for our prime number cruncher benchmark program (IA Jun 80): 3,300 seconds.

Two versions of Pascal are available. One is the familiar UCSD Pascal to execute the P-code of any program developed for a UCSD-compatible computer. The other is Pascal/MT, a subset of UCSD Pascal running under the normal operating system for 88. As a result, the data files created in Pascal/MT are compatible with those used by a Basic program. Pascal/MT programs are assembled right down to 8080 machine code, making them execute very fast. A version of our prime number cruncher zipped through in 119 seconds.

Macro-88 is included for 8080 assembly-language programmers, including macro definition capabilities and other tools that the factory has developed for its own use while creating the operating system.

Finally, Digital Research's CP/M is now available on all Polymorphic computers. The consumer should be able to purchase any of the thousands of canned CP/M-compatible applications programs being offered.

Varied applications programs

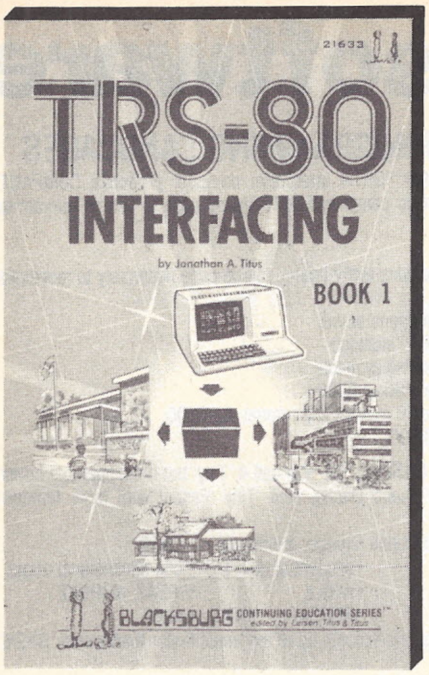
The applications software catalog is thin. It includes a manufacturer's inventory package and mailing list program, both in daily use at the company's factory. Also listed is a financial modeling tool, Plan, a "what if?" information handling program sharing some of the features of Visicalc by Personal Software. Also, there is a listing of about 50 programs available from local dealers.

We witnessed a preview of Datamaster, a forthcoming language/applications package. It will be a database management system combined with a super high level applications program development tool. It will allow a non-programmer to make intelligent inquiries into a database and create immediate customized management reports. It is being translated from Basic into Pascal/MT.

A Twinsystem 88 with a pair of double-sided, double-density 5¼-inch diskette drives lists for $13,200. This includes two terminals, interconnecting cables, diskettes containing the operating system, editor, Basic, and the macro assembler and user documentation. It also includes a set of comprehensive confidence test programs. Two double-sided, double-density 8-inch diskette drives cost an additional $4,885. Software pricing ranges from $95 for the mailing list program to $195 for Wordmaster II.

It's feasible to purchase a single-user, single-diskette system 8810 for $5,208 and build it up to the full capabilities of the Twinsystem 88 later. □

NECESSITIES FOR THE COMPUTER USER



TRS-80 Interfacing Book I
By Jonathan A. Titus
Order No. 10058
192 pages $8.95

If you have a fairly good understanding of the commands in Level II BASIC, this book will appeal to you. The book introduces you to the signals available within the TRS-80 computer and shows you how they can be used to control external devices.

Computer Dictionary
By Donald D. Spencer
Order No. 19010
160 pages $5.95

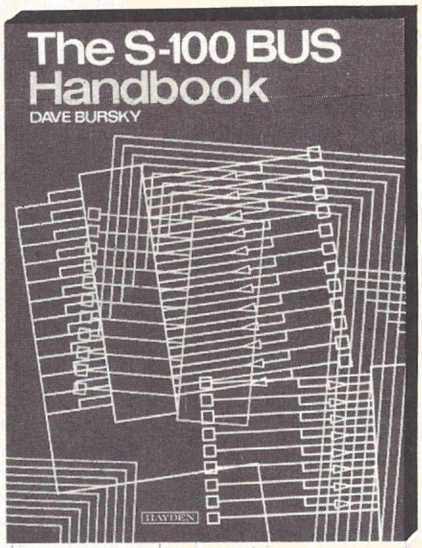
This current and comprehensive dictionary contains about 2,500 words, phrases, and acronyms used in connection with computers. The keynote of this book is clarity and precision without sacrifice of authority. All definitions are simple, and stand as independent units of explanation.

BASIC Programmer Primer
By Mitchell Waite and Michael Pardee
Order No. 10032
240 pages $8.95

This will serve as an invaluable tool to anyone who wants to learn BASIC, the most popular computer language of today. Seven chapters explain the fundamentals of BASIC, program control, organization and a game program, additional functions, and variations of BASIC.

Some Common BASIC Programs
By Lon Poole and Mary Borchers
Order No. 21001
200 pages $9.50

This book contains program listings and documentation for 76 short programs covering financial, mathematical, statistical and general interest applications. Every program has been tested; examples and sample program runs are published with the source listings for each program.



S-100 Bus Handbook
By David Bursky
Order No. 16033
280 pages $12.95

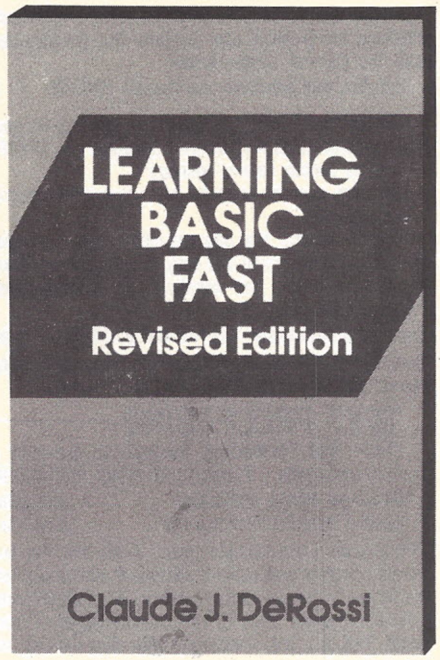
Computer fundamentals, basic electronics, and the parts of the computer are covered here. It explains all the operating details of commonly available S-100 systems. Schematic drawings of all the boards discussed in the manual are provided for reference. Each of the major system boards is discussed in complete detail as to how it operates and how it connects into the rest of the system.

Programming & Interfacing the 6502, With Experiments
By Marvin L. De Jong
Order No. 10059
448 pages $13.95

Filled with information and programs, this will be useful to anyone interested in using 6502-based microcomputer systems.

How to Program Microcomputers
By William Barden, Jr.
Order No. 10009
256 pages $8.95

This book explains assembly language programming of microcomputers. Basic concepts, including number systems and operations, basic computer operation, and computer codes are examined. Microcomputer architecture and operation are also discussed.



Learning BASIC Fast
By Claude Derossi
Order No. 20011
211 pages $11.95

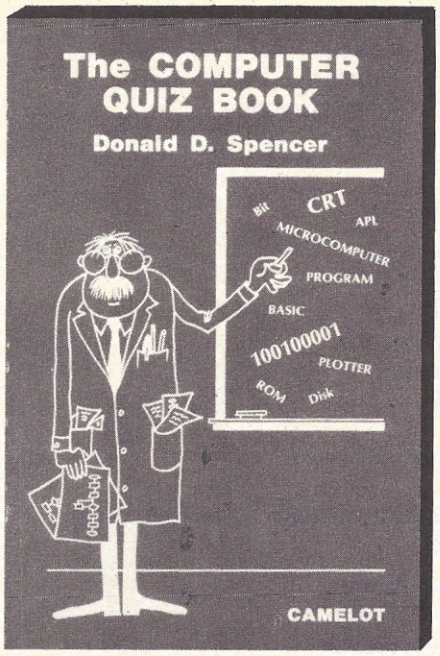
Provides simple, concise explanations of computer systems, flowcharting, and the important commands used in Basic. Requires no high-level math. Readers can begin programming almost immediately.

Peanut Butter and Jelly Guide to Computers
By Jerry Willis
Order No. 14022
225 pages $7.95

This delightful book begins with a series of case studies of different people who have units up and running. Readers are led through the essential information needed to become informed users of personal computers. The book is a combination consumers guide, tutorial and offers some thoughts on where you should buy your computer.

DATA DYNAMICS TECHNOLOGY

has the books to make it easy



The Computer Quiz Book
By Donald D. Spencer
Order No. 19009
128 pages $5.95

This book is written for readers who would like to test themselves on basic computer concepts. It may be used effectively by students, teachers, laymen, programmers, personal computer users, or anyone else interested in checking their knowledge of computer concepts.

From Dits to Bits
By Herman Lukoff
Order No. 14018
200 pages $12.95

Follow along as one of the pioneer developers of the computer recounts the intense atmosphere of discovery and frustration that pervaded the back room of Moore School at the University of Pennsylvania. The text is illustrated with numerous photographs that effectively recreate the atmosphere of the late 40's and early 50's. A glossary has been provided for the newcomer to the world of computers.

Build Your Own Working Robot
By David L. Heiserman
Order No. 13013
234 pages $5.95

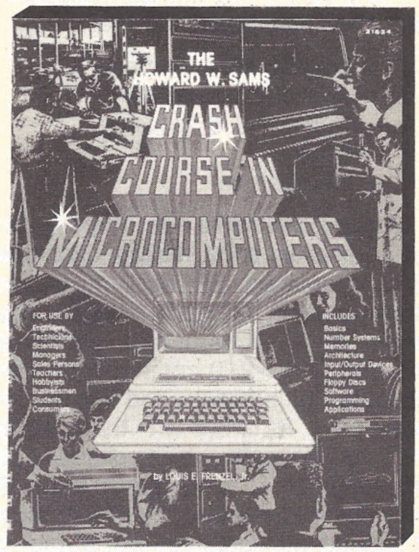
Here are complete instructions, plans, schematics, logic circuits, and wiring diagrams for building Buster, the mechanical pet robot. He'll serve you coffee or bring you the morning paper. He'll forage for his own "food" and scream when he can't find it. His "curiosity" will get him into one plight after another, but Buster has the capacity to get himself out of trouble just as easily as he got into it!

How To Profit From Your Personal Computer: Professional, Business, and Home Applications
By T. G. Lewis
Order No. 16007
208 pages $9.65

Written for business people and computer hobbyists, this book shows you how to put your computer to work for you. It describes the uses of personal computers in common business applications, such as accounting, handling payrolls, managing inventory, sorting mailing lists, and many others. Many BASIC language programs are reproduced to illustrate the techniques for putting your computer to work for you.

The Howard W. Sams Crash Course in Microcomputers
By Louis E. Frenzel, Jr.
Order No. 10053
264 pages $17.50

Now you can stop struggling with conventional theory/reference-type computer books. The special "crash" course is organized into a series of 14 easy-to-learn units. This unique self-taught text gives you the most for your money and is a complete, fully-illustrated learning resource on the world of microcomputers.



10/80

DATA DYNAMICS TECHNOLOGY P.O. Box 1217, Cerritos, CA 90701

Name (Print) ____________________

Address ____________________

City ____________________ State ______ Zip ______

Please send me:

Order #	Qty	Price	Total	Order #	Qty	Price	Total

Shipping & Handling Charges $.75 ea. U.S., $1.50 ea. Foreign

TOTAL ORDER $________

***TAX $**________

SHIPPING & HANDLING $________

TOTAL ENCLOSED $________

☐ master charge ☐ AMERICAN EXPRESS ☐ VISA

____________________ ☐ Check or M.O.(U.S. Funds drawn on U.S. bank)

Exp. Date ________ Signature ____________________

*California residents add 6% sales tax. Availability and prices quoted subject to change without notice.
Please allow six weeks for delivery. You may photocopy this page if you wish to keep your INTERFACE AGE intact.
Orders *cannot* be shipped unless accompanied by payment, including shipping & handling and tax where applicable.

DATA DYNAMICS TECHNOLOGY, A Division of INTERFACE AGE Magazine (213) 926-9548

Place a Special Order on Real Estate

by Gary A. Stotts

The Multiple Listing Service System is designed to aid the real estate agent in matching real estate listings to buyer requirements. The agent can enter the number of bedrooms desired, the price ranges, and the location desired, and the system will list all real estate listings that match those requirements. Designed for the Apple microcomputer with one to three disk drives, and an optional printer, the MLS System features multiple volume files, linked lists and a free storage list.

The system must be initialized once using the MLS Index Create program. The number of disk drives available is entered. The number of MLS index entries is set at 1,144, and can be calculated by multiplying the number of locations times the number of price ranges times the number of bedroom options: locations × 11 price ranges × 4 bedroom options ÷ 1,144. Each index entry consists of an identifier for the attribute combination this index entry references, and a pointer to the first record with this attribute combination.

The program will first ask if printed output is desired. Next the menu screen prints:

1 — Lookup MLS property
2 — Add an MLS property
3 — Sell an MLS property
4 — Stop
5 — Reorganize

Option 2 will add a new property to the system. Attributes (location, price, number of bedrooms) are prompted for and entered. Further information concerning the property such as MLS number, square feet, selling terms, style, construction and heating method are entered. Line 1210 computes an identifier, I1, for this attribute combination, and line 1220 computes which index entry, R, to use with these attributes. Lines 1250 and 1260 compute which volume to place the new property listing on. I1 is compared to the Rth index identifier array element. If equal, the new listing is added to a chain of listings with common attributes; if zero, this is the first listing with this attribute combination, if not equal, add 1 to R and check the next index array element.

Option 1 will look up all MLS listings with common attributes. The attributes are prompted for and entered. Each listing is then individually displayed. Processing similar to option 1 occurs to find the correct index entry. The index entry points to the address of the first record with the required attributes, which in turn contains the address of the next record with these attributes, and so on.

Option 3 will update an MLS property listing with a status code. An 'E' for earnest money, 'S' for sold, or 'D' for deleted can be entered.

Option 5 will remove any records marked as 'D' or 'S' and reorganize the file. Removed records will have their address stored in a free storage list so that new property listings can make use of the space. All links around the deleted record are updated to reflect the change. □

MLS Index Record Layout

DS — number of index entries
NXT — address of next record to write
ND — number of disk drives
FS$ — free storage list
DI% — attribute index
DP% — pointer index

MLS Data Record Layout

Record Code: A — Active
D — Deleted
E — Earnest money
S — Sold
MLS number
Property address
Property price
Number of bedrooms
Number of bathrooms
Terms: 1 — Cash
2 — Conventional
3 — Assume
4 — FHA
5 — VA
6 — Owner will carry
Square feet
Style code — construction code — heating code
Link to next record

Style Codes: 1 — Ranch
2 — Bilevel
3 — Trilevel
4 — Two story
5 — Bungalow
6 — Chalet
7 — Mobile home
8 — A-frame
9 — Other

Construction Codes: 1 — Frame
2 — Brick
3 — Frame/brick
4 — Stucco
5 — Moss rock
6 — Adobe
7 — Log
8 — Modular
9 — Other

Heating Codes: 1 — Gas
2 — Electric
3 — Propane
4 — Solar
5 — Wood
6 — Coal
7 — Oil
8 — Other

Program on Page 128

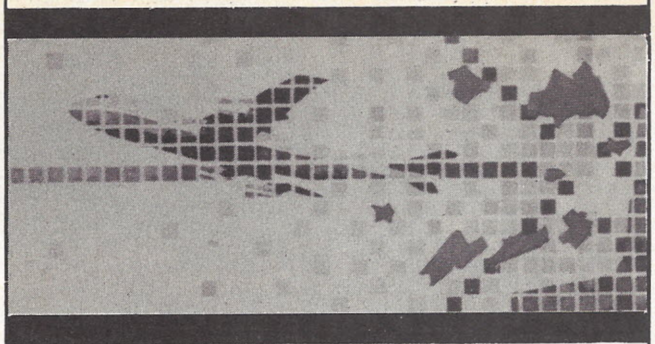


Table of variables for the computerized flight simulation by Thomas Carbone (IA, Aug 80)

```
Variables:

 A- Acceleration (Ft./Sec.)
A$- Gate sign graphics
A1- Auto pilot (1=On, 2=Off)
 B- Brake (0=Off, 4=On)
B5- Resultant bearing
 C- Radio (1=On, 0=Off)
C1- Input command
C3- Brake failure
 D- Heading (Degrees)
D5- Distance travelled
 E- Reserve brake
E(- Engine thrust (0=Off, 4=Full)
E1- Throttle direction
F(- Fire warning
 G- Time (GMT)
G$- Select graphics mode
G8- Flight progress
 H- Horizon angle
H(- Hydrolic pressure warning
H8- Quadrant sign change
H9- Ground wind speed
 I- Gear position (1=Up, 0=Down)
 K- Brake pressure (Lbs./In↑2)
K3- Reserve brake pressure
K9- Taxi distance
 L- Altitude (Feet)
 N- Longitude position
N$- Select normal mode
O(- Oil pressure warning
O9- Total thrust
 P- Flap angle (Degrees)
Q9- Fault engine
 R- Radio frequency
R$- Select reverse video mode
R7- Flight path
R8- Tower radio frequency
R9- Radio transmission select
 S- Air speed (Mph)
S1- Stick position
 T- Latitude position
T1- Elapsed time between commands (Sec.)
T3- Maximum thrust of engines
T8- Angle of elevation (Degrees)
T9- Angle of elevation (Radians)
 U- Fuel (Lbs.)
U9- Aircraft progress marker
 V- Vertical speed (Ft./Min.)
V8- Distance travelled on Bradley runway
V9- Distance travelled on Kennedy runway
 W- Window heat (1=On, 0=Off)
W(- Water flow warning
W2- Change in bearing
W3- Wind direction re: aircraft
W5- Sign of W6
W6- Distance travelled in X direction
W7- Distance travelled in Y direction
W8- Wind speed
W9- Wind direction
 X- Row position
 Y- Column position
 Z- General purpose
Z$- Printing string
Z1- General purpose
Z2- Variable to be printed
```

INTRODUCING T.I.M.

You've decided that a computer might help you perform certain functions in your business more efficiently. Now you are looking at different machines and software. Here's where your problem starts.

You can compare disks against disks and printers against printers. But when it comes to software there are so many programs on the market, it's like comparing apples to oranges. If you buy the right one you win. If you don't, you lose.

That's where we come in. To help you win. We're Innovative Software and we've just finished developing and field-testing a new program for the micro-computer market. It's called T.I.M., Total Information Management. We've been told that it's one of the most comprehensive data base management programs on the market. Check out these features and see if you agree:

- Menu-driven and user-oriented. Extensive computer expertise is not needed.
- Contains 4 user libraries. Any number of file formats, report formats, list formats, and record selection formats may be stored.
- Contains a unique text-editor that may incorporate field information from the data base to write checks, invoices, form letters, etc.
- Eliminates resorting of data files by automatically merging new or updated records back into the host file.
- Supplied with a 125-page reference manual with over 30 pages of examples.
- Available under Micropolis BASIC and CP/M* MBASIC**

See how T.I.M. can help you win in the software game. If your dealer doesn't have T.I.M. yet, give us a call or write to the address below.

Innovative Software
Computing Power at Your Fingertips

9805 Holly
Kansas City, Mo. 64114
(816) 942-8778

*Trademark of Digital Research
**Trademark of Microsoft

MICROS HIT THE JACKPOT

by Michael Panchak

Back in 1973, the late Howard Hughes—a mean protector of the coin of the realm, especially his own—discovered that his gaming operations in Las Vegas were being ripped off. The bandits each had one overworked arm, and Hughes knew exactly who they were . . . along with all their accomplices. He also discovered that he was forced to pay for each bandit's personal upkeep at the oft-recurring rate of $14 per hour, and that the bandits numbered in the hundreds. To make matters worse, he didn't know how much they were skimming off, or when and how they would strike. And no law enforcement agency was the least bit sympathetic.

Hughes instructed his company, Summa Corp, to find someone who could come up with a cheat-proof slot; one that would baffle the casino operators, managers, mechanics and Vegas street characters whose scams were ingenious, uncountable . . . and robbing him of 10% of the nightly take.

Summa's search led to Interscience Systems, whose president, James Halverson, recognized the possibilities of converting an electromechanical slot with a microprocessor. He formed Summit Industries in Reno, Nev. and a year later came up with just such a machine. Hughes died in the interim, and interest in the cheat-proof slot went with him.

But a good idea dies hard. Halverson reintroduced it 2 years ago after some design modifications to the Nevada Gaming Commission, whose approval is a must for any legal gambling device. That was won in January 1980 and ushered in the slot of the future. And what a future.

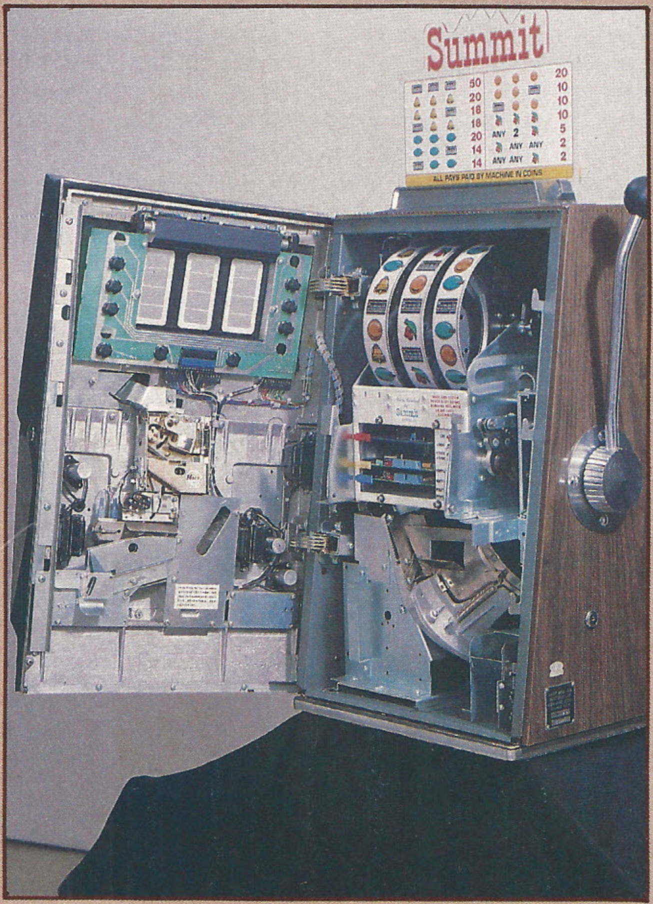
According to Andy Lucero, Summit's development director and chief designer of the Autoslot package, there are some 85,000 slot machines in Nevada alone. He has an eye to convert them all; but at present has a target of about 25% . . . "a ton of machines" as he likes to put it. It's also a ton of money—about $30-$40 million—and this doesn't include the action just booming on the Jersey coast or that sure to come in New York and Miami.

Besides stumping the cheats, two other problems had to be overcome in the micro-based slots: auditing the payouts and maintaining true randomness.

Every coin that drops in a slot, theoretically, must be counted and reported to the Nevada commission. But the drop-in counter can easily malfunction in the old-style electromechanical machines, perhaps missing as many as every third coin. That can tally into a sizeable dollar amount being lifted without anyone the wiser. As Lucero puts it: "The machines flucuate wildly, usually above their theoretical hold (the theoretical hold signifies optimum operation—paying out 85% of the

Standing side-by-side among the thousands of slots in a typical Vegas casino, this Summit Corp. micro model is innocently indistinguishable from the electromechanical types. But its micro-based interior has irradicated a host of other problems that once plagued casino operators, especially in its two most troublesome areas: security and maintenance. Electromechanical slots, with their array of stepping motors, lots of contacts, relays and stepping switches, have to be constantly maintained . . . at a wopping cost of $14 per hour. And even with the most elaborate maintenance program, the typical machine gets the heck beat out of it in a 24-hour period. Malfunctioning is a constant problem. And no player is going to inform the casino when a machine is repeatedly paying off or hitting jackpots. In places like Atlantic City, just introduced to casino gambling, the problem is compounded by the high humidity of the ocean resort. Jersey is already in the process of going completely micro.

WITH THE CHEAT-PROOF SLOT



The interior of a converted slot. The center card cage contains microprocessor board, interface board and communications board. One of the chief features of micro adaptation is that it eliminates three of the commonest player scams: the "sick-machine cheat" who can spot a malfunctioning slot after four or five handle pulls and gets it to "rhythm" jackpots—legally—because the machine is not selecting stops randomly due to improper lubrication or neglected maintenance; the shimmy expert who, with a piece of super-thin shim stock, available at any auto parts shop, can find a hair-breadth aperture in the outer case, maneuver his shim over the bracketry, and stop the free-spinning wheels at will. His grandmother can then come along, claim the jackpot and walk out; or the cheat with a 20,000 rpm motor flexible carbide drill bit literally up his sleeve, who can bore right throught the tough metal casing (installed, incidentally, just to keep him out), reset the payout mechanism on the hopper and quietly sit there collecting coins.

coins dropped and holding 15% for the house).

"Everyone says 'casinos get more money than they should.' But that's not the point. Gambling is a business. If you have a machine that is supposed to pay out 85% and hold back 15%, you want to see that 15% in your pocket. If you can't predict what you'll get, you have a problem." The micro-based machine stifles all fears. It will truncate to the exact theoretical hold until it falls apart in a heap of dust.

But if payouts have to be highly predictable, the opposite is true of the machine's "play." It must have total randomness. Every player should have the same probability of hitting a jackpot or earning a score with each handle pull as any other.

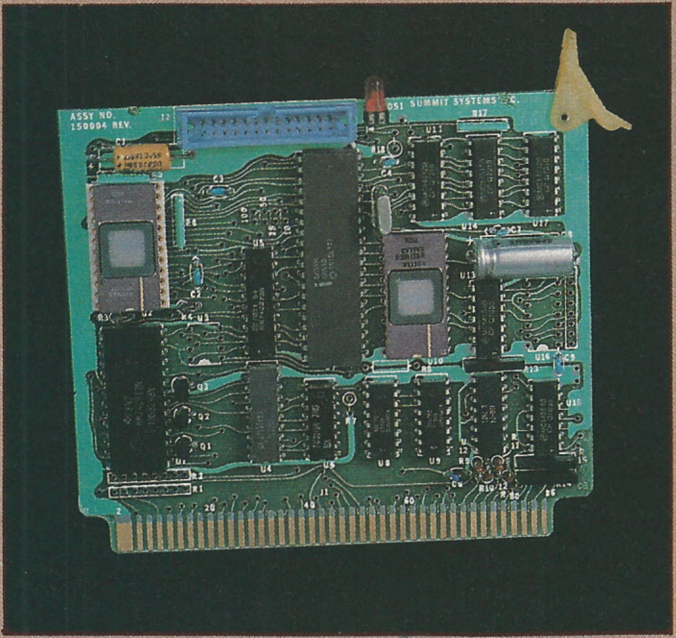
The typical slot has three wheels with 22 stops each. That's 22^3 or 10,646 possible combinations. To find out if a machine is truncating to its theoretical hold takes 10 cycles of operation—or over 100,000 handle pulls. It can take a mighty long time . . . and cost a good deal of money to find out if randomness is lacking and the machine is truncating at all possible stops. (The opposite can also occur: the machine is truly random, but pays out a higher amount than it should.) The micro completely randomizes and, in the event of over or under payment, picks it up instantly.

The mechanics of gambling aside, the microprocessors also had to emulate the feel of the old mechanical devices. "Slots have an 80-year history," Lucero points out. "They bring back an automatic—but not electronic—era." The slots from that period had a certain wobbly feel, handle pulls made a distinctive, unmistakable sound that "talked" to the player, the spinning wheels gyrated off-sync. The micros reproduce these qualities perfectly. More important, they still give the player the feel that he has some control over handle pulls and kickbacks.

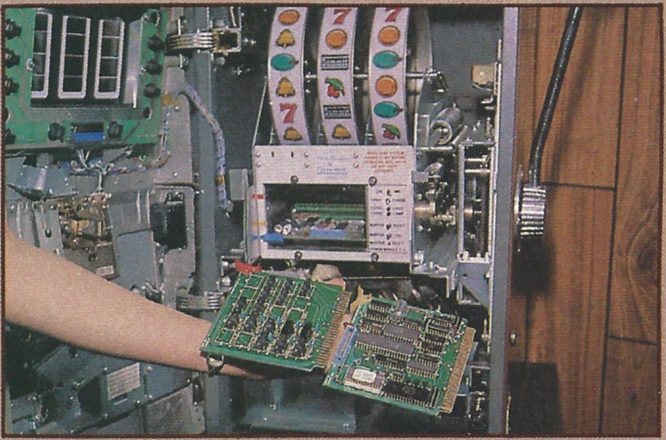
"People talk to them: 'come on baby, let's have it' or 'I've got this turkey now'. I know an old grandmother in southern California who insists that she has three slots in the Dunes Hotel that talk just to her. They don't, of course. They are purely random devices. But we don't tell her that."

Lucero forsees a time when the various state gaming commissions will demand third party audits, wherein they can enter into a node unbeknownst to the casino operator and gather their own raw data on the casino's operations right from the floor. When it comes, his "converts" will be ready.

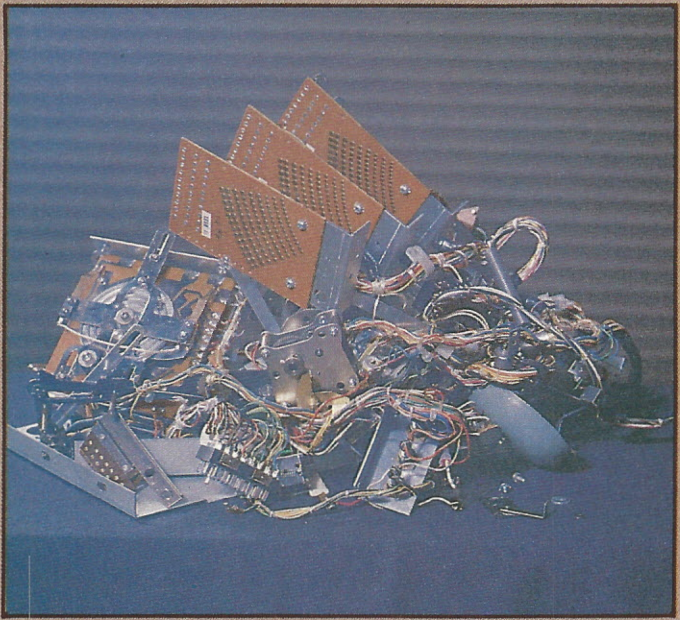
Howard Hughes progeny will be in good stead. And Lucero as chief beneficiary smiles quietly at the bright, flashing future ahead.



Main CPU board and interface board control all the functions of the slot machine . . . including cost accounting. The micro-controlled conversion package with nonvolatile memory can store a vast amount of data: number of coins in, number out, number dropped, handle pulls, number of coins in the hopper, jackpots, refills, whether the machine was entered and what was done to it. It can be hooked up to form its own database, enabling any bit of information to be retrieved days after its occurrence. A node can be inserted between the slot and the CPU, tied into a CRT, screen monitored instantly, stored and "called back" days later. A small printed circuit card is inserted and hooked up to the CPU for hourly, daily, monthly and yearly financial reporting. In short—a stand-alone machine that is very secure, cannot be set up, does not have miles of wires and switches that can be manipulated to pay out erroneously, has no problem coming out of adjustment, and does not rely on mechanical devices that have to be properly lubed up to prevent rhythming and cockeyed randomness.



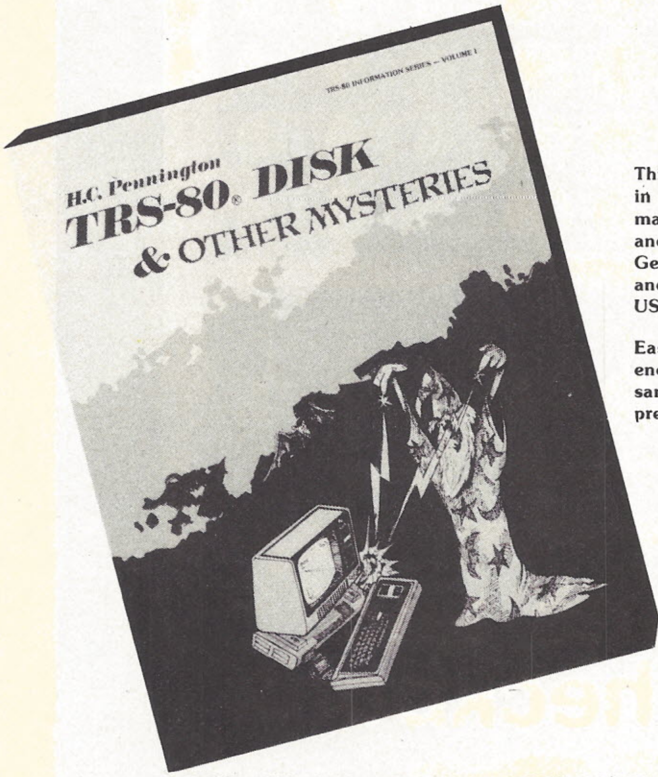
Only two boards are required to control all the functions of the slot. A third board provides communications to a CPU for security and eliminates the old-fashion in-house scam, say a slot mechanic with a few inside accomplices who opens a slot allegedly to fix a contact, shorts it out, and closes the machine. A cohort comes along, plays the machine until it's empty and signals for a fill. The mechanic makes the fill, undoes the contact, and no one ever suspects that the machine was bilked. The variety of ways the electromechanical contacts can be adjusted probably number in the hundreds. A good slot cheat —that is a shady mechanic and some secretive accomplices—can swing a ton of money out of a casino.



Some of the approximately 45 pounds of electromechanical parts gutted from a standard slot during conversion. They make up more than 90% of the internal controls, and are the parts most apt to malfunction or permit cheating. Onconversion costs $1,500-$2,000; 95% of slots now in casinos can be switched over on the spot in about 2 hours per machine. After installation, the system operates on its own—easily manageable by the slot department without the need for skilled technicians or any changeover in personnel. Systems can have customized software that reports any special data the casino operator wants, i.e., how many machines were opened more than twice in the past 24 hours.

THE **HOTTEST**

"How To Fix It . . ." EVER WRITTEN!!

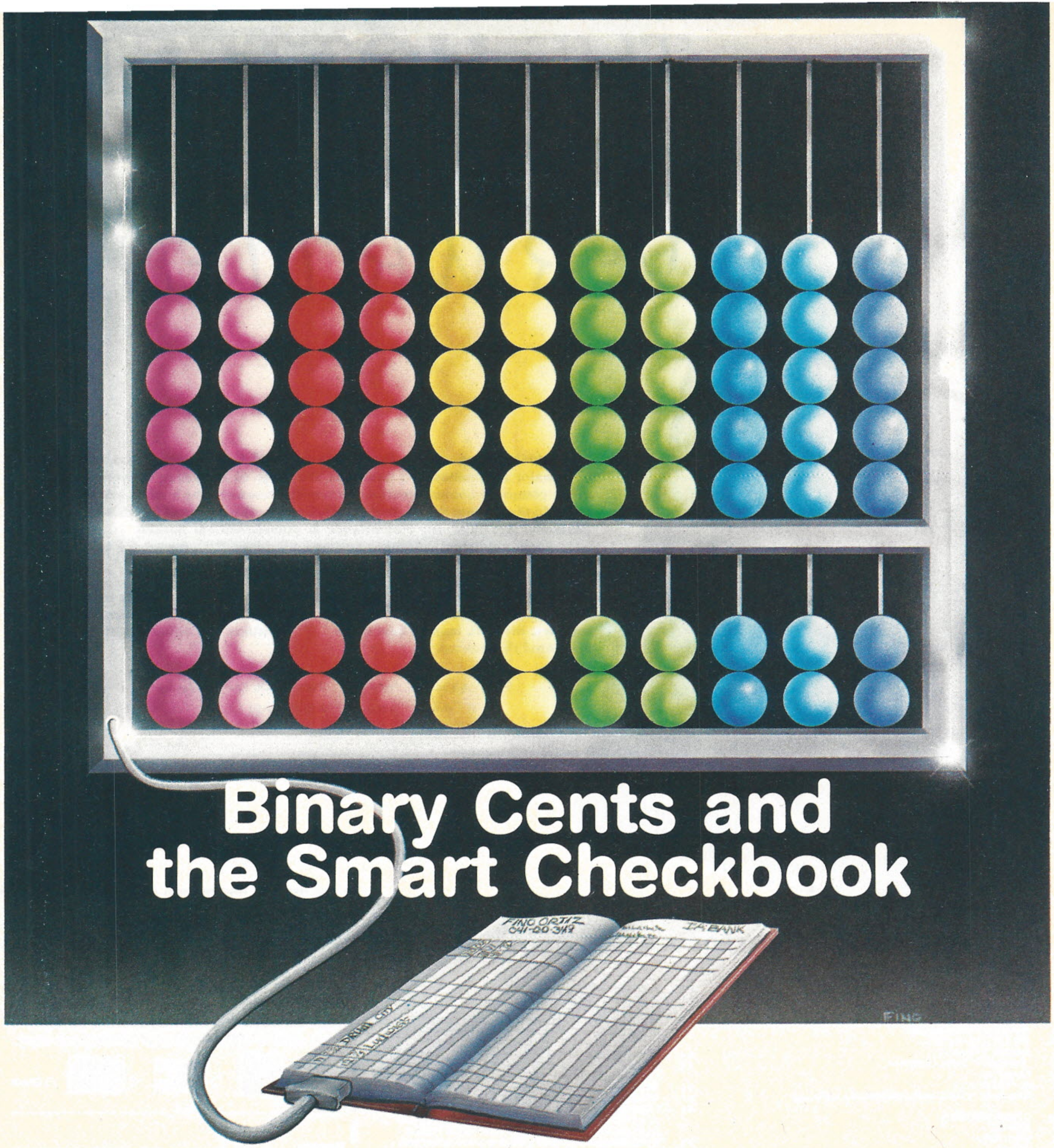


This valuable textbook, the first in a series by H. C. Pennington, has been reviewed in 3 major magazines and has been the topic of extensive conversation on two major computer communications networks. Presently, the book is being distributed and sold throughout the United States, Canada, the Union of South Africa, England, Germany, Australia, the Netherlands & others. William Barden, Jr., a well known and widely published computer author comments, ". . . my advice is to get it and USE it!"

Easy to read, entertaining material covers nearly every aspect of the many problems encountered on the TRS-80 system with disk storage. Actual examples, problem samples, memory maps, "how to fix . . ." solutions and countless illustrations are presented throughout.

Author: H. C. Pennington   Publisher: IJG Computer Services
First Edition - Second Printing - Soft Cover - 130 Page
8½" x 11 — Wt. 1 lb. — Retail price: $22.50

TRS-80 DISK & OTHER MYSTERIES TABLE OF CONTENTS



Binary Cents and the Smart Checkbook

by Murray L. Lesser

There is some justification in the charge that you are wasting time if the only application for your super-whiz home computer (aside from games) is balancing your checkbook. However, many banks now give out information they never handed to customers before, and one needs computation ability to make the most of it. My bank will tell me what they think my daily balance is any time I ask by sticking a card into the slot in their 24-hour terminal. While I don't quite trust their system design enough to transact any business through the terminal that involves passing money back and forth, I am quite willing to use the information they are giving me to know when my checks clear.

The trick is to find the subset of the "not cleared" checks that I have written. This sum will equal the disbursement the bank claims has been made since the last time I ran the program. One could try all possible combinations, but this seems a little crude. I use a tree pruning algorithm that is more elegant. Before we get to the algorithm, let us look at those difficulties in writing checkbook programs that are created by high-level language implementers who insist that computers should do binary arithmetic when solving decimal problems.

Binary in the beginning

Ever since John Backus delivered the first successful high-level language compiler—Fortran—in 1957, the implementers of high-level languages seem to have believed that the innards of computing engines have never changed. Backus' system took in decimal num-

bers and converted them to binary. Fortran did all the arithmetic in binary, and then converted the stuff back to decimal for output. It didn't take long for smart users to discover the booby trap built into the procedure—conditional branches did not always occur as expected by those born with ten fingers and toes.

Twenty-five years later, RAM now costs about ⅕¢ per bit retail, and there is hardly a computer alive that can't do decimal arithmetic almost as easily as binary. But language implementers hardly ever learn. My new Microsoft Basic-80, version 5, still does binary arithmetic deep down inside, and I still have to do unnatural things to my programs to make them find the right answer.

Allegedly there are a few Basics that do arithmetic directly in decimal. Yours probably works in binary if you have ever been surprised by getting something that looked like "-5.58794E-08" on a printout when common sense told you that the answer should have been zero. There are some simple tests you can run to check for binary Basic. One is the listing in figure 1. This program is in ANSI minimal Basic, and should run as written on any machine that claims its Basic is Basic. Only the results will vary. If you have true decimal arithmetic, there will be no third column in the output. My Basic produced figure 2.

If you have 'print using' or equivalent in your Basic, you can control the printouts. But you have to use subterfuge to assure that your program knows what to do when your bank balance reaches real zero. Some neophyte programmers get so confused with what their binary Basics are doing that they give up.

The difficulty stems from the fact that binary and decimal are two different number systems, and the twain will never (well, hardly ever) meet after the decimal (or binary) point in a finite register length machine. The only time you can be sure that the value of the variable in binary Basic will actually be equal to what you think it is (unless you think in binary) is if you are dealing with whole numbers.

Once you understand this simple fact, you never again need to be troubled by imaginary bugs and inconsistencies when making "equals comparisons (=)." All you have to remember is that your software is probably quite reliable at doing what its implementer thought it should do. But, this may not be what you think it does. To get these two thoughts together, all you have to do is to make all your comparisons between pairs of whole numbers that lie entirely within the significance range of the Basic you are using.

Such comparisons are easy in dollars and cents computing, because you can do all comparisons in pennies and know that you are not losing any significance. In the program shown, I use the penny function, as defined in line 560, in all value comparisons in order to prevent a few stray bits at the bottom of a register from throwing me into the wrong loop.

Locating the uncleared checks

The algorithm (subroutine starting at line 4000) used to find which checks might have cleared is very simple. We have the total amount the bank says has been paid, and we have an ordered list of the outstanding checks. We start with that total, let's call it temp, and the value of all the outstanding checks, which we will call remainder. We look at each check in sequence. If temp is greater than or equal to the amount of the check, we post the check number ('checkno') to a list of possible candidates, and subtract amount from temp. If the amount is greater than temp, we know that this check doesn't belong in our solution, as found to date, so we don't post it. In either case, we subtract amount from remainder and move on to the next check in the list.

Eventually, one of two conditions will arise. Temp may go to real zero. That means that we have found a solution to the problem. More likely, initially, remainder will decrease to a value less than temp, and we know that we are on the wrong branch of our implicit tree.

When remainder drops to less than temp, we back up to the last node in the tree for which we posted a check, and prune at that node. The amount of the most recently posted checks is added back to temp, and a new remainder is calculated for all of the checks in the list following the one we just pruned. We then proceed down the new branch, until either of the two conditions arises again.

If we back ourselves completely off the tree, pruning every possible branch, there is no solution and something is wrong with the input data. Maybe the bank socked us with a service charge that we forgot to take into account. Or maybe the bank credited us with a little interest. More likely, we keyed in a wrong balance or check amount.

Breaking the tie

One important consideration: there may be more than one solution to the mathematical problem our algorithm is solving. That is where the tie breaker

```
A>TYPE ITEST.BAS
  00100        FOR I = 1 TO 10
  00200        LET A = 1 + I/100
  00300        LET B = A - 1
  00400        PRINT A, B,
  00500        IF B - I/100 = 0 GOTO 700
  00600        PRINT B - I/100,
  00700        PRINT
  00800        NEXT I
  00900        END
```

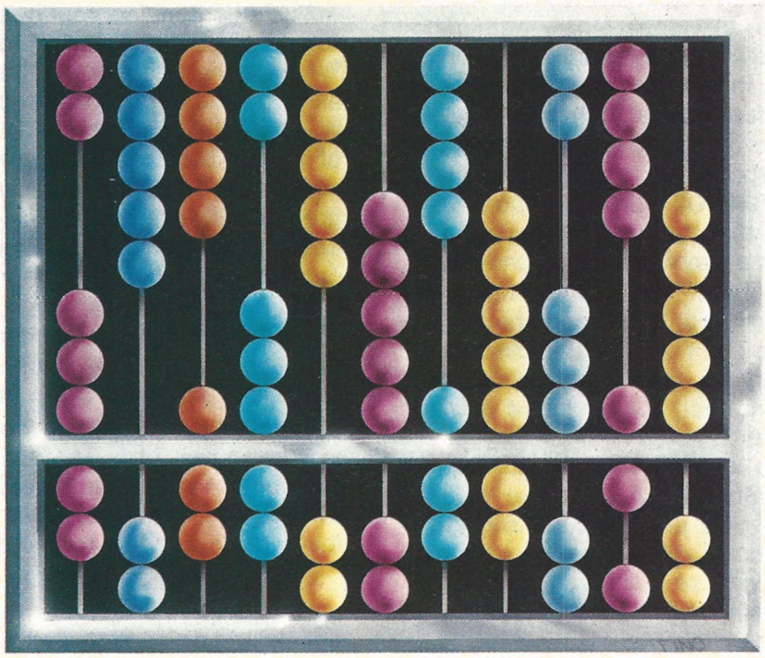
Figure 1. Test for binary zero.

```
A>ITEST
 1.01          9.99999E-03   -9.31323E-09
 1.02          .02           -1.86265E-08
 1.03          .03           -2.79397E-08
 1.04          .04           -3.72529E-08
 1.05          .05           -4.84288E-08
 1.06          .0599999      -5.58794E-08
 1.07          .0700001       5.21541E-08
 1.08          .0800001       4.47035E-08
 1.09          .09            3.72529E-08
 1.1           .1             2.23517E-08
```

Figure 2. Sample output of test.

comes in. Suppose you had two $50 checks and one $100 check outstanding, and the bank says it paid $100 on your account. Not even your super-whiz home computer is psychic. So the tie breaker in this algorithm is the order in which the checks appear on the list. If at least one of the $50 checks appeared on the list before the $100 one, the solution given would be that the two $50 checks had cleared.

The degenerate case for the algorithm is if the only check cleared was the last one on the list, and its amount was more than the sum of all the others that precede it. The program will then try every possible branch all the way to the bottom before it finds the answer. You can speed up the solution by sorting the checks into decreasing amount order before running the tree. But this will change the tie breaker—the $100 check will always be earlier on the list. I order my data in check number sequence, so my tie breaker is first in, first out. Sometimes I have special knowledge, so I cheat and adjust the input record. It is always easier to use the program to find out if there is more than one possible solution, but you need special knowledge to know which solution is right.



Programming information

The program shown is written in Microsoft's Basic-80, version 5. Since mine is the compilable one (Bascom 5.01), I have been lavish with remarks and lengthy mnemonic variable names in the listing—neither of which are carried over into the machine-language run-time version. If you are using an interpretive Basic and are tight for space, you will find that pruning the variable names down to their first two characters will not introduce any ambiguities. It will also prevent the REM in remainder from causing difficulties in versions of Basic that don't allow buried reserved words in variable names.

If you are not familiar with Basic-80, version 5, there are some other changes from earlier versions you should be aware of before you read the program listing. One is the elimination of the need for semicolons as concatenating separators in variable lists in 'print' or 'lprint' statements—spaces are acceptable substitutes (e.g., line 1060).

'While...wend' has been added to satisfy the anti-'goto' structured programming bigots. If you don't have 'while...wend,' you can substitute an 'if...goto' for the 'while' (reversing the sense of the logic statement) and a 'goto' for the 'wend.' For example, line 1120 would become 'if eof(1) goto 1170,' and line 1160 would be 'goto 1120.'

Another substantive change in version 5: Microsoft has belatedly recognized that 'for...to' loops should always be tested at the 'for...to' line, not at the 'next' line. This means that if the index already exceeds the 'to' value on entry, there will not be an unwanted first pass. If your Basic is deficient in this respect, you will have to fix it with a detouring 'if...goto' line before entering the loop that starts on line 5030. Such a detour is not needed before line 4180, because the circumstances can never arise at that point.

Line 4220 is put in because I hate to stare at a blank display screen wondering if anything is going on inside the machine. The pruned check numbers will be displayed on each pass. Unless you scroll off by spending too much time climbing too many branches, the complete path will remain on the face of the screen when the program finishes—providing you sent the output to the printer as shown.

The full-blown Ascii listing shown (written with Microsoft Edit-80) is stored on 8K of disk. The final operational machine-language program lives on 19K of disk, but that includes all the system functions it needs that otherwise would be provided by the Basic interpreter. I don't know the minimum memory size in which this program would run in interpretive mode. I am certain that a properly transliterated version would fit easily in a system that has the Basic interpreter in ROM and 16K of RAM. It might even run in 8K. Since the file used is sequential, it could be on a cassette tape instead of on disk as shown.

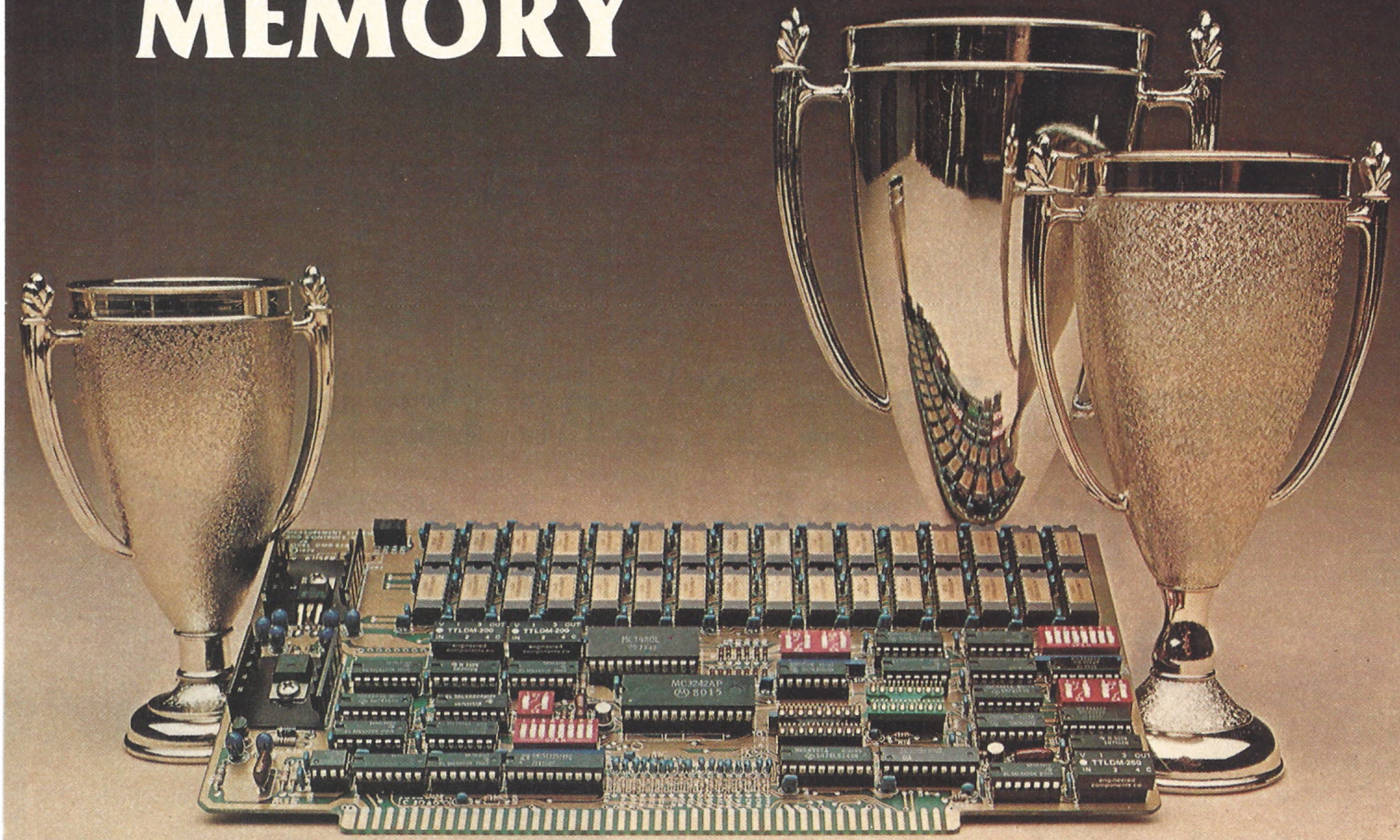
Carrying along the 'payee' in the records is of marginal utility—the added information is barely worth the effort in keying it in. On the other hand, if you are inserting the algorithm into an already existent check register program, you can embellish it by adding another array variable to carry the check date.

A bug creeps in

I used this program in this form almost daily for over two months, and it hasn't blown yet. That is not to say that it is bug free. In fact, a subtle bug has shown up. It is in Microsoft's release 5.01 of their Bascom. The penny function FNA(W), defined in line 560, will produce single-precision results, in spite of the fact that the <name> of the function, A, has been defined as a double-precision variable. This bug has allegedly been fixed in release 5.1. Test your version of Basic, and, if necessary, insert a 'defdbl f' in the program to fix this implementation bug.

I have used a variety of programming styles, as an exercise and would suggest that you study the listing until you understand the concepts, and then reprogram the application to fit your own environment. □

THE UNBEATABLE S-100 MEMORY



That's the MEASUREMENT systems & controls DMB Series of S-100 bus memory modules, fully compatible with **ALPHA MICRO, CROMEMCO, NORTH STAR, MP/M,** and most other S-100 systems.

Definitely a winner, the DMB Series is available with Bank Select (DMB6400) or without (DM6400) and utilizes industrial quality construction, provides outstanding reliability, and is backed by dedicated customer service and a one year guarantee.

The DMB6400 uses I/O port addressing for the bank select feature. A switch provides the ability to select any one of the 256 I/O ports for addressing the memory banks. The memory is configured as four totally independent 16K software selectable banks, with each bank addressable on any 16K boundary.

Outstanding features such as those listed below make the DMB series the UNBEATABLE S-100 Memor,.

- Four independent 16K software selectable banks.
- Each bank is independently addressable on any 16K boundary.
- Switch selectable bank sizes — from 16K to 64K in 16K increments.
- Eight banks (512K) per I/O port for each of the 256 ports.
- Z-80 4MHz operation with no wait states using transparent refresh.
- On-board diagnostic LED's.
- Low power — 8 watts maximum.
- Reliable, tested and burned-in memory.
- IEEE S-100 compatible timing.
- One year guarantee.
- Attractive Dealer & OEM Prices.

See your nearest computer dealer, or contact us for the complete story on the UNBEATABLE S-100 Memory.

Systems Group

A Division of MEASUREMENT systems & controls incorporated

867 North Main St. / Orange, Calif. 92668 / (714) 633-4460
TWX/TELEX: 678 401 TAB IRIN

Program Your Phone Calls to Dial...Time... Rate...Re-call

by Fred Blechman

If you own a microcomputer, you can communicate with the world by just adding a $3 relay. Sound far-fetched? This Basic program, written for the Radio Shack TRS-80 Level II but adaptable to other micros, will hold up to 500 names and telephone numbers in a 16K RAM memory (about 40 names even with only a 4K memory). It can also be used to call the same number over and over.

When you type in the name of the party you wish to call, the computer sequentially searches in memory for a match. When it's located, the screen displays the name and phone number, including area code, then pulse-dials the number, displaying each digit as it "dials."

You can command a total listing on the screen, or manually dial from the keyboard, or redial the last number dialed, with a single key (plus 'enter'). During the call, an on-screen elapsed timer tells you the number of minutes charged and how many seconds remain for the next additional charge. At the end of the call, you can enter the rates for that particular call (varying with distance and time of day) and the computer calculates and displays the total charge, and asks for the next command.

Screen prompts make the program easy to use, and no knowledge of programming is required if


1
DEF
3
GHI
4
MNO
6
PRS
7
8
WXY
9
*
OPER
0
#
282-7650
LEE '80

you have a TRS-80 level II. For other computers, some timing and pulsing commands may have to be changed.

Simple and inexpensive interface

The big bugaboo for most nonelectronic types would normally be the telephone interface. However, the interface in its simplest form (see figure) is just a common Radio Shack relay, light-emitting diode (LED) and 9-volt transistor radio battery connected to the recorder cable and one wire of your telephone line. Even a novice in electronics could make this hook-up for you. The extra plug, jack and switch shown can be packaged in a small box with the relay, LED and battery to eliminate cable switching from computer to dialer mode.

The connection to the phone line is accomplished by merely opening one phone wire (red and green are the commonest colors) at the instrument or the wall connection or anywhere in between. This open connection is completed by the normally-closed relay contacts. Although this series-type connection is not a safety hazard, cannot disable the phone lines, and is not detectable by the phone company, most telephone company regulations require that you notify them before attaching anything to the phone lines.

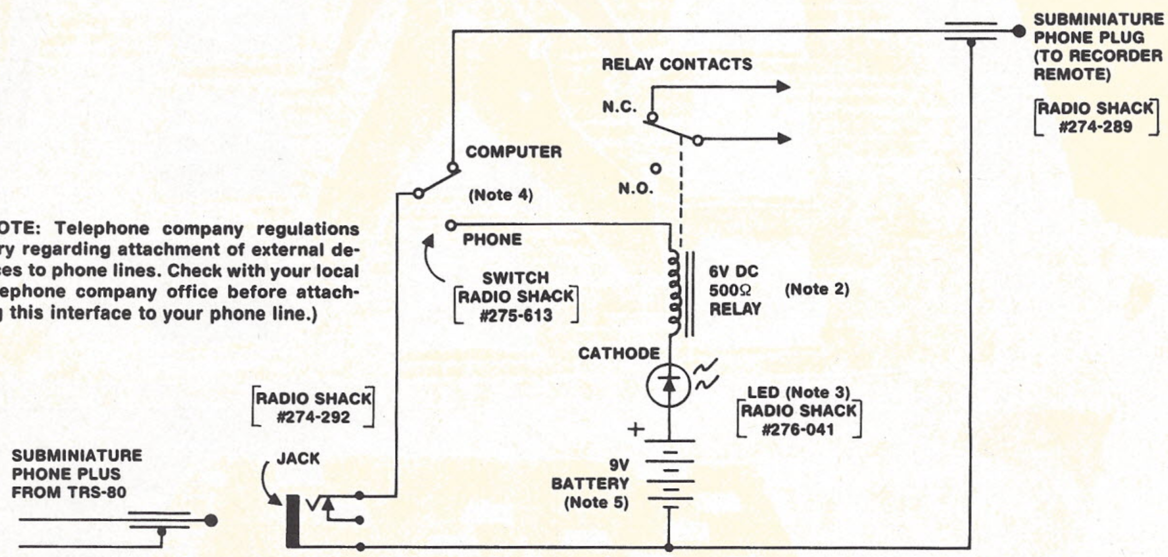
Notice that this program requires no modification to the computer itself. Using the external relay protects the contact of the cassette recorder motor-control relay in the TRS-80 keyboard unit.

Inserting phone numbers

When you enter the program from the keyboard, use your own phone number list for the data statements. The screen listing will be in the same sequence as the data statements, so if you want an alphabetical screen display, enter your data lines in name-alphabetical order. Each data line contains the name, local phone number and area code, separated by commas. Don't use any dashes between numbers. If the phone number is in your own area code, enter a zero. If an area code is used and you normally access this with a 1 from your phone, add a '1' before the area code in the data statement.

This dialing and interface system may be used with all U.S. single-party telephone circuits, rotary or tone. However, some old systems may not be able to accept the fast dialing pulses timed into this program. If that seems to be a problem, change the 50 in line 590 to 200 and the 10 in lines 1020 and 1030 to 15. (Line 590 controls the pulse interval while lines 1020 and 1030 control the pulse duration.)

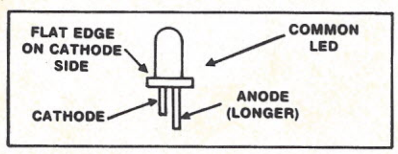
The elapsed time function is controlled by line 780. The value of 266 should work well at normal room



(NOTE: Telephone company regulations vary regarding attachment of external devices to phone lines. Check with your local telephone company office before attaching this interface to your phone line.)

NOTES

1. Normally-closed relay contacts. Break phone line (red wire) and connect with contacts in series. Pulsing relay opens circuit and generates dialing pulses. Circuit must be closed for normal phone operation.
2. Radio Shack #275-004 or equivalent.
3. LED performs several functions:
   (A) Visible evidence of dialing pulses
   (B) Reduces battery drain
   (C) Warns if switch in wrong position for computer operation (LED lights when TRS-80 commands recorder on)
4. Switch allows you to select dialer or regular computer operation without moving cables.
5. 2U6 9-volt battery or 6 pencells in series.



Telephone Dialer Interface Schematic

temperature. If your unit runs too fast or too slow, change 266 to a higher or lower number accordingly. Run the program from the beginning for each timing test, since apparently the variable table in memory is scanned with each loop of line 780. Testing with a shortcut approach like 'goto 700' or 'run 700' will give a false (faster) count.

The actual programming approach used here could be greatly improved with some commands available in TRS-80 level II Basic, but perhaps not available in other Basics. This approach uses common Basic commands to minimize language compatibility problems. However, the dialing and elapsed-timer functions need special attention. The TRS-80 has a relay within the keyboard unit that is operated by two Basic commands: 'out255,4' closes the normally-open relay contacts (which operates the added external relay, opening its normally-closed contacts), and 'out255,0' opens the relay contacts (which allows the external relay contacts to return to the normally-closed condition). These commands are found in lines 1020 and 1030, together with the 'for-next' timing loops that control the pulse interval and duration. Other computers will need appropriate port commands to energize the external relay. The relay itself may need to have a higher or lower coil operating voltage, depending on the port voltage.

Other computers will need appropriate port commands to energize the external relay...which may need a higher or lower coil voltage.

All the 'for-next' timing loops (lines 590, 780, 1020 and 1030) will probably require higher numbers to maintain the same real-time duration for other micros, which mostly run faster than the standard-speed TRS-80.

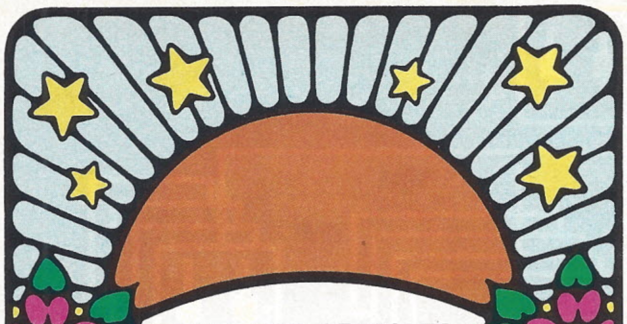
Dealing with limitations

The three limitations on the total of names and numbers retained in data statements are RAM memory available, loading time and search time. Each data line uses about 25 bytes of RAM.

With 67 names and numbers in memory, it takes 86 seconds to load this program from a cassette at the normal level II 500 baud rate. A speeded-up TRS-80 could take as little as 43 seconds. Using an Exatron stringy floppy, running at its regular 7200 baud, this same 67-name program loads in only six seconds. Put the same program on disk and it will probably load in two or three seconds.

Once the program is loaded, run time when searching for a particular name depends on how far down the data list the desired name resides. Using 67 names, it takes about three seconds for the TRS-80 (normal speed) to find the last name on the list. Obviously, the last name on a 500-name list in a 16K memory could take over 20 seconds. You might devise alphabet-jump routines to shorten this search time, but I didn't find it necessary. □

Program on Page 132



WIN!

more than 1000 sweepstakes prizes for the video lifestyle of the '80s.

from **Home Video**

the one magazine that reflects, shapes and guides the video lifestyle.

2 Grand Prizes	Complete Magnavox Home Video System: VCR, Color Camera and Color TV
2 1st Prizes	TheaterVision Big Screen TV
1 2nd Prize	Magnavox Portable VCR
1 3rd Prize	Magnavision Videodisc Player from Magnavox
1 4th Prize	Magnavox Color Video Camera
178 5th Prizes	Top Movies and Specials on Cassette from Magnetic Video, Allied Artists, Nostalgia Merchant, ABC, Time-Life and Video Tape Network
125 6th Prizes	Maxell Blank Videocassette
200 7th Prizes	Shape Videobox cassette storage unit
600 8th Prizes	Fotomat discount for conversions to tape

Bring home all the excitement of home video every month with a subscription to Home Video. You'll find previews of programs on cable and pay-tv, and reviews of the latest cassettes and discs. You'll read all about the new equipment — VCR's, cameras, discs, big screen tv, computers and games — what they do, how they work, and how they can provide entertainment, education and creativity for you and your family. Make the most of the new home video lifestyle — subscribe to Home Video today.

OFFICIAL RULES — NO PURCHASE NECESSARY

1. Complete the official entry blank or on a 3″ × 5″ piece of paper, print your name, address and zip code. Enter as often as you wish. Mail each entry separately to: Home Video Magazine Sweepstakes, P.O. Box 101, Church Street Station, New York, N.Y. 10046. Entries must be received by November 30, 1980.
2. Winners will be selected in random drawings conducted by The Independent Judging Organization, Inc., whose decisions are final. All prizes will be awarded and winners notified by mail. Only one prize to a family. Winners may be required to execute an affidavit of eligibility. Prizes are not exchangeable for cash. Liability for taxes is the responsibility of winners.
3. Sweepstakes open to U.S. residents except employees and their families of Home Video Magazine, its printers, publishers, dealers and I.M. Towers & Company, Inc. Void where prohibited.
4. For a list of major winners, send a stamped self-addressed envelope to: Home Video Winners List, P.O. Box 134, Church Street Station, New York, N.Y. 10046.

Sign up for the money-saving subscription now. Enter sweepstakes.

Mail entry to: **HOME VIDEO MAGAZINE SWEEPSTAKES**
P.O. Box 101, Church Street Station
New York, N.Y. 10046

☐ YES Enter my subscription to HOME VIDEO MAGAZINE and enter me in the HOME VIDEO SWEEPSTAKES.

For	Bill Me	Payment Enclosed
1 Year	☐ $13	☐ $12
2 Years	☐ $21	☐ $20

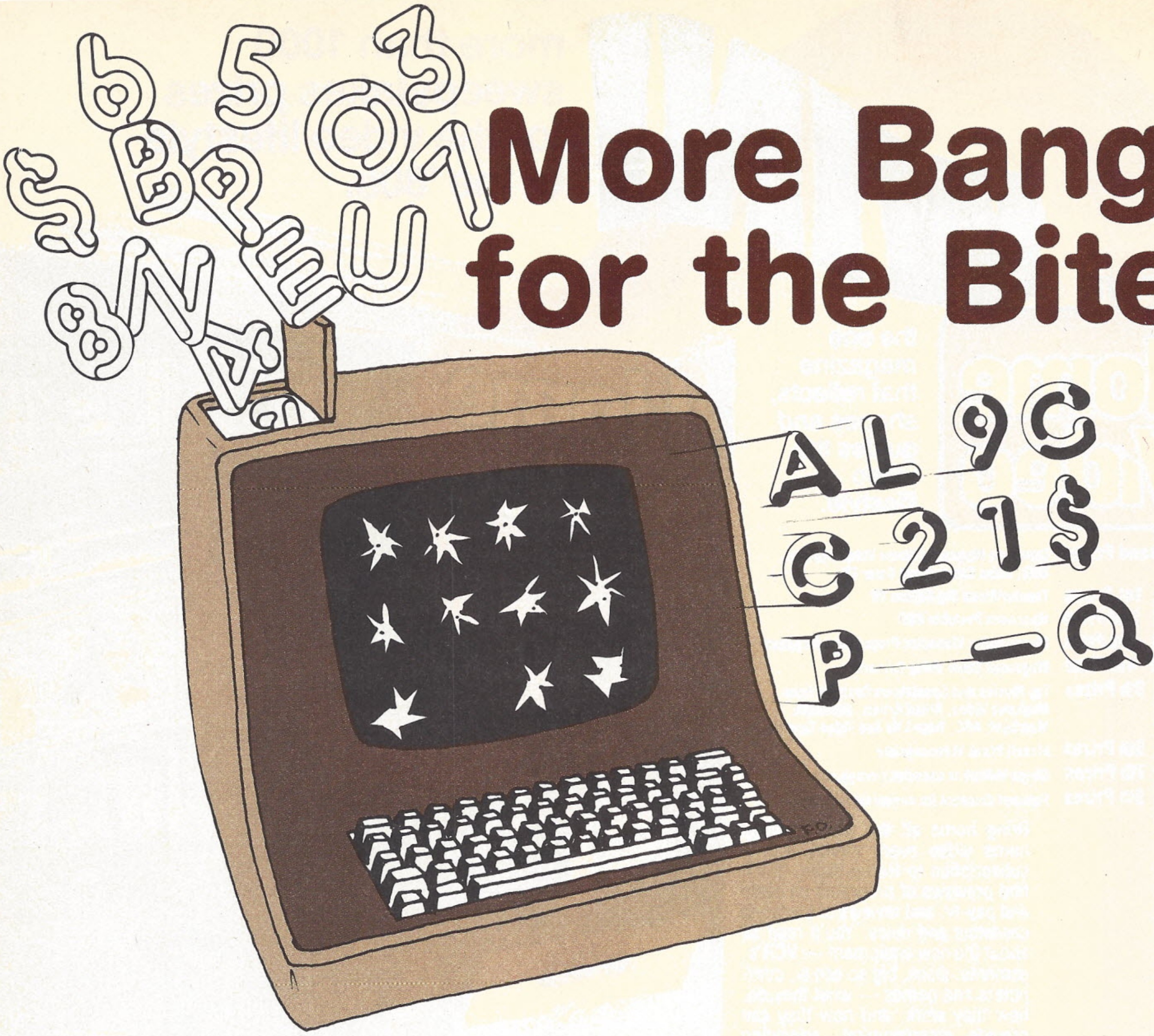
☐ YES I'm already a subscriber, please enter me in the sweepstakes and extend my subscription as indicated above.

☐ NO Don't enter my subscription now, but enter me in the HOME VIDEO SWEEPSTAKES anyway.

Name ____________________

Address ____________________

City ____________ State ____________ Zip ____________



More Bang for the Bite

by David Veldof

Do your data structures suffer from that full, overstuffed feeling? Have you ever wished you had 24K of memory when you only had 16K? Then, this simple remedy might be worth a try.

It's a method of representing a subset of the Ascii characters in Base 40 notation. Base 40 makes possible storing three characters in 16 bits (2 bytes) instead of the normal two Ascii characters. The storage capacity of all data structures holding Ascii information is increased by 50%. For example, if your program devotes 2000 bytes of storage to hold 2000 Ascii characters, using Base 40 will allow you to store 3000 characters in that same 2000 bytes. An additional plus is that this also decreases read/write times of peripherals such as cassette or disc since data is condensed by 50%.

For those in the crowd who are one step ahead, yes, there is a drawback. Base 40 represents only 40 characters out of the full Ascii complement. The characters normally represented in Base 40 are the upper-case alphabet (A through Z), the digits 0 through 9, and four other special characters, usually the dollar sign, space, period and dash (see figure 1). In practice any 40 different characters can be represented by the Base 40 method. You can substitute the lower-case alphabet for the upper case or substitute three more

Base 40 Character Set

Ascii Character	Base 40 Equivalent
Space	0
A - Z	1 - 26
$	27
.	28
-	29
0 - 9	30 - 39

Figure 1. Characters normally represented in Base 40.

special characters for the letters X through Z, etc. My discussion, however, will use the character set defined in figure 1.

Actually when you think about it, the normal Base 40 character is pretty comprehensive. It includes enough characters to represent names and numerical data, plus—for you bottom line people—the ever-popular dollar sign. Armed with this basic representational set, it should be possible to store just about anything.

Now that you know that Base 40's "drawback" won't hamper your style, let's take a look at how the characters are stored. To approach this properly, you must first understand fully the way in which the alphanumeric characters are stored in Ascii format. Each character (7 bits) fits very nicely into one byte of storage and that's that, no more and no less. This makes life simple for the programmer who is coding the string-handling routines for the Basic interpreter. Micros are basically byte-oriented so it's easy to address one byte out of a group or to move one byte to another, etc. On the other hand, life is a little more difficult for the programmer trying to stuff 3K of characters into a 2K bag.

Storing several letters

When two or more Ascii characters are stored, they are saved in continuous bytes in memory. If you look at the storage requirement for two Ascii characters, you find it is 2 bytes. Two bytes will hereinafter be referred to as a "word." If you count the number of bits in a word, it will be 16 (see figure 2.) The bits are numbered 0 through 15 starting with bit 0 on the right (the lower order) and ending at bit 15 on the left (the high order).

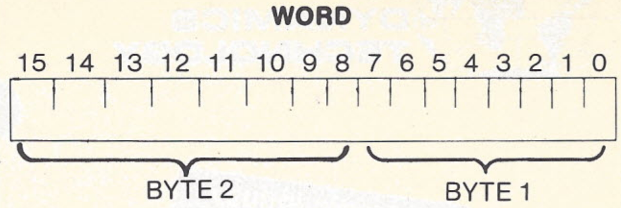


Figure 2. Number of bits in one word.

"Big deal," you say. All that's been done is to fit Ascii characters in 2 consecutive bytes in the computer storage. But let's look at it from another angle. In Basic, integers are represented using 2 bytes (one word). All operations involving integers (i.e., adds, subtracts, divides, etc.) involve using the entire word when performing the computation. With Ascii characters, however, a word contains two distinct characters each aligned on its own 8-bit boundary. Operations performed on one Ascii character of the word (i.e., Boolean operations) will not affect the other character.

In short, Ascii is stored logically and integers are stored arithmetically in the word.

Base 40 is stored like an integer is in Basic. Let's see why. The name Base 40 implies that 40 distinct characters (decimal codes 0-39) can be defined. If Base 40 took the Ascii approach to saving data, it would take 6 bits (39 decimal is 100111 binary) to define one Base 40 character. A little arithmetic will show that you would only be able to store two Base 40 characters (with 4 bits left over) in one computer word. Not exactly the best deal you ever heard of, right? OK, let's look at the way it's really represented.

Since we can store three Base 40 characters in 16 bits, let's arbitrarily give the characters the names "H", "M", and "L." H is for the high order, M for the middle and L for the low order. The characters will be stored in the word HML. Next, let's look at the packing algorithm used to get HML into the word. It is:

HML = H*40**2 + M*40 + L

* = multiply

**2 = raise to the second power

Fooling around with this expression, we can rewrite it as:

((H*40) + M)*40 + L

Overcoming computer inadequacy

Now we have it where we want it. Almost. Notice the two multiplication operations. As we all know by now, microcomputers are rather deficient in the multiplication department. They can add and subtract just fine; but try and get them to multiply and they clam up and play dumb. Fine, you say, looks like we can't use that algorithm. But wait, the number 40 can be expressed as (32 + 8). If we were to replace 40 with 32 + 8 in the previous expression we'd have:

(HM*32 + HM*8) + L

where:

HM = ((H*32 + H*8) + M)

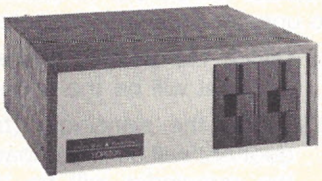
You might wonder how we've improved the expression with multipliers in it by substituting one with four

NO FRILLS! NO GIMMICKS! JUST GREAT

DISCOUNTS

MAIL ORDER ONLY

ATARI 800
Personal Computer System
$799.00



NORTHSTAR

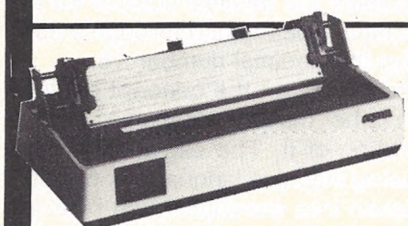
Horizon II 32K	$2349.00
Horizon II Quad	2799.00
Horizon II 64K	2999.00
Horizon Quad 64K	3399.00

TELEVIDEO

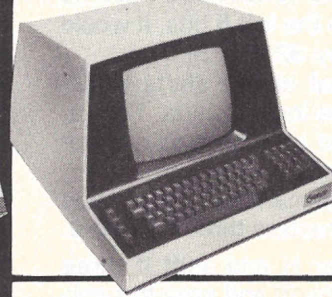
912 **$749.00** | 920 **$799.00**

HAZELTINE

1420	$795.00
1500	$849.00
1510	$1049.00
1520	$1229.00

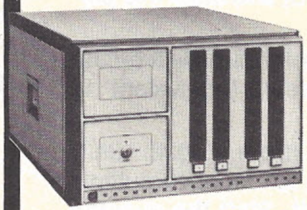


OKIDATA
Microline 80
$699.00



SOROC Technology

IQ 120	$699.00
IQ 140	999.00

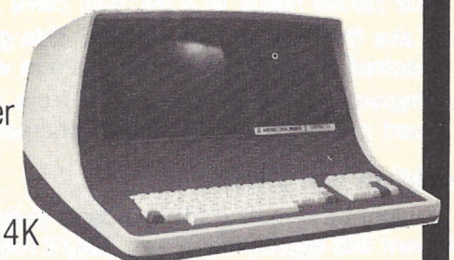


CROMEMCO

System 3	$5695.00
Z2H	7995.00

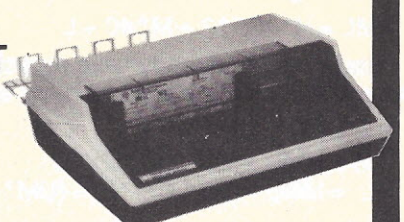
INTERTEC
Superbrain
32K Computer
$2495.00

Superbrain 64K
$2795.00



DECwriter IV
LA 34 **$979.00**

TEXAS INSTRUMENT
810 Multi Copy
Impact Printer
$1499.00



We'll meet or beat any advertised prices!

Most items in stock for immediate delivery. Factory-sealed cartons.
Full manufacturer's guarantee.

DATA DISCOUNT CENTER

Box 100 135-53 Northern Blvd., Flushing, New York 11354
Visa • Master Charge • N.Y.S. residents add appropriate Sales Tax • Shipping F.O.B. N.Y.

PHONE ORDERS CALL
212
465-6609

CIRCLE INQUIRY NO. 70

multipliers. The answer lies in the fact that the coefficients 32 and 8 can be expressed in powers of 2. 32 is 2^5 and 8 is 2^3. Getting back to our truant micro, we find that it can really multiply after all if you only want to multiply by powers of 2. It is possible to shift data in the computer any number of bits to the left. Each left shift is equivalent to multiplying the quantity by a power of 2. So, if we were to take the Base 40 character called H and shift it to the left 5 bits, it would be equivalent to multiplying it by 32 (2**5).

It should be remembered that all left shifts to accomplish multiplication should be logical shifts. That is, the sign bit (bit 15) of the word is treated as a bit of magnitude not as an algebraic sign.

The following section summarizes the steps necessary to pack Base 40 characters three to a word.

1. Take the Base 40 character H and shift it three places to the left (multiply it by 8) and save this partial result.
2. Shift H two more places to the left (multiply it by 32) and add the saved partial result (equivalent to H*40). Save this as the sum.
3. Take M and add it to the sum (from step 2). Shift this quantity over three places to the left (multiply by 8) and save the partial result. Shift the quantity two more places to the left (multiply by 32) and add the partial result to it (M*40). Save this in sum.
4. Take the sum and add L to it. This gives you three packed Base 40 characters in one word.
5. Repeat the whole procedure from step 1 for the next group of three Base 40 characters.

Avoiding one pitfall

It might be tempting for you Basic purists to try to implement this algorithm in Basic. You'd run into a problem though because Base 40 representations of characters having the decimal code 21 or greater will cause an 'arithmetic overflow' error on statements such as this one from Radio Shack level II Basic:

```
10 REM Pack High Order Base 40 character "U"
20 HO% = 21%*40%*40%
```

% = defines a constant or variable as a one-word integer

Now let's attack the reverse problem, unpacking Base 40 characters. Remember the original equation for packing:

HML = H*40**2 + M*40 + L

Assuming you already have a packed Base 40 word, the equations for unpacking it are as follows:

UH = HML/40**2
UM = HML – (UH*40**2)/40
UL = HML – (UH*40**2) – (UM*40)

where:

UH = unpacked Base 40 high order
UM = unpacked Base 40 middle order
UL = unpacked Base 40 low order

There's not much we can do about the divides by 40 or 40^2 except implement them using a software divide subroutine. The other expressions aren't really as formidable to compute as they look. For example, the first equation:

UH = HML/40**2

when computed, will give a quotient equal to the character desired (i.e., UH) and a remainder that can be expressed mathematically as:

Remainder = (HML – (UH*40**2))

This is exactly what is required on the right side of the equation to compute UM. Similarly, the result of the computation:

UM = (HML – (UH*40**2))/40

will give a quotient of UM and a remainder that can be expressed mathematically as:

Remainder = HML – (UH*40**2) – (UM*40)

This is what is needed on the right side of the equation for UL. In other words, when UM is computed, the remainder will be UL. To summarize the steps necessary to unpack Base 40 characters:

1. Divide the packed characters (HML) by 40^2. The quotient will be the unpacked high order.
2. Divide the remainder (from step 1) by 40. The quotient will be the unpacked middle character and the remainder will be the unpacked lower-order character.

We have now shown how to pack and unpack Ascii characters, assuming they are already expressed in their Base 40 equivalents. But how do we get them in their Base 40 equivalents?

Converting those extra symbols

Looking at figure 1, we see that we can make special compares for the Ascii characters space, dollar sign, period and dash and convert them immediately to their Base 40 equivalents. The Ascii characters A through Z can be detected in a similar fashion and their Base 40 equivalents can be computed by subtracting the Ascii character by one less than the Ascii codes for the letter A. For example, assume the character to be converted to Base 40 is the letter C (decimal representation 67). Subtract the decimal number 64 (i.e., one less than the decimal representation of A) from it to obtain 3, which is the Base 40 equivalent. The conversion of the digits 0 through 9 proceeds in an analogous fashion.

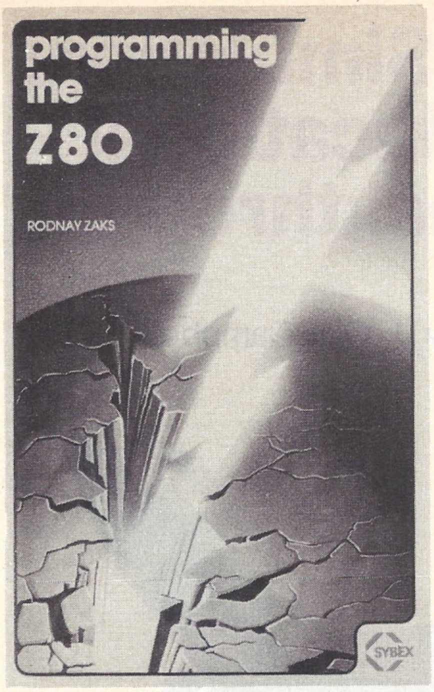
To convert from Base 40 to Ascii, use the same procedures as Ascii to Base 40 translation. Detect the Base 40 characters space, dollar sign, period and dash by exact compares. The letters A through Z can be converted to Ascii by adding the decimal number 64 to the Base 40 representation. The digits 0 through 9 can be converted by adding the decimal number 18.

We have concentrated more on the concept behind Base 40 representation rather than the mechanics of any computer's implementation of it. The explanation has been slanted toward using machine language however. This will have to be taken into account if you plan to write Base 40 routines in some upper-level language.

The Base 40 concept has been around for a long time. This method of character storage is normally used by programmers developing assemblers and compilers in order to store information in symbol tables. Sometimes it is used to represent names of files on a disc. Anytime that space is at a premium and the range of characters to be represented is limited, is an opportunity to use Base 40. □

Books for Your System from

DATA DYNAMICS TECHNOLOGY
A division of INTERFACE AGE Magazine
Box 1234, Cerritos, California 90701



Programming the Z-80
By Rodnay Zaks
Order No. 0-89588-013-X
620 pages $14.95

Covers the essential aspects of programming as well as the advantages and disadvantages of the Z-80, and brings the reader to where he/she can write complete application programs.

Z-80 Microprocessor Programming & Interfacing: Book 1
By Elizabeth A. Nichols, Joseph C. Nichols, and Peter R. Rony
Order No. 0-672-21610-18
496 pages $12.95

The second of two laboratory-oriented texts, addresses the interfacing of digital circuits.

8080 Machine Language Programming For Beginners
By Ron Santore
Order No. 0-918398-14-2
104 pages $6.95

This beginners' book will take you step-by-step through the most common 8080 op codes at a level everyone can understand.

8080/8085 Software Design: Book 1
By David G. Larson, Peter Rony, Jonathan A. Titus, and Christopher A. Titus
Order No. 0-672-21541-1
336 pages $9.50

A detailed approach to assembly language programming for the 8080 or 8085 based computers.

The 8080A Bugbook: Microcomputer Interfacing and Programming
By Peter R. Rony,-David G. Larsen, and Jonathan A. Titus
Order No. 0-672-21447-4
416 pages $10.50

The principles, concepts, and applications of an 8-bit microcomputer based on the 8080 microprocessor IC chip, with emphasis on the computer as the controller.

6502 Software Design
By Leo Scanlon
Order No. 0-672-21656-6
288 pages $10.50

This material is presented to increase the reader's understanding of the 6502. Fundamentals are first explained then more complex topics are gradually introduced.

DATA DYNAMICS TECHNOLOGY P.O. Box 1234, Cerritos, CA 90701 **10/80**

Name (Print)__________
Address__________
City __________ State______ Zip______

Please send me:

Description	Qty	Price	Total	Book Order #	Qty	Price	Total

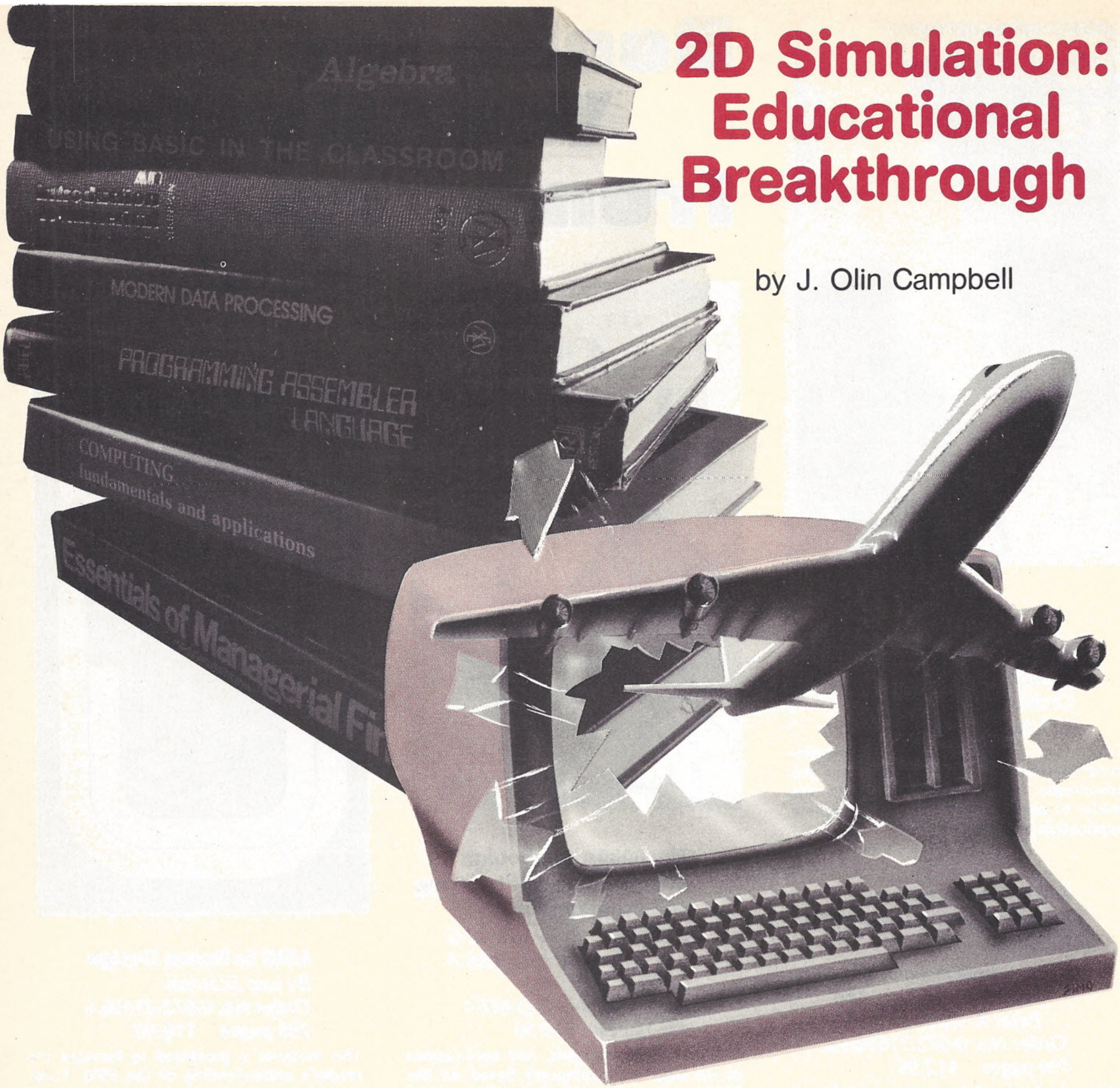
Shipping & Handling Charges **TOTAL ORDER $**______
***TAX $**______
Books $ .75 ea. U.S., $1.50 ea. Foreign **SHIPPING & HANDLING $**______
TOTAL ENCLOSED $______

☐ master charge ☐ American Express ☐ VISA # ______ ☐ Check or M.O.(U.S. Funds drawn on U.S. bank)
Exp. Date______ Signature ______

For European delivery contact **INTERFACE AGE Europe**, Dahlienstr. 4, D-8011 Munchen-Vaterstetten, West Germany.
*California residents add 6% sales tax. Availability and prices quoted subject to change without notice.
Please allow six weeks for delivery. You may photocopy this page if you wish to keep your INTERFACE AGE intact.
Orders *cannot* be shipped unless accompanied by payment, including shipping & handling and tax where applicable.
DATA DYNAMICS TECHNOLOGY, A Division of INTERFACE AGE Magazine (213) 926-9548

2D Simulation: Educational Breakthrough

by J. Olin Campbell



Our present education delivery system is reaching the limits of its improvability. The last decade has shown that pouring money and effort into our present tools scarcely improves the output. There is little reason to believe that more of the same will be any more effective. Our teachers need better tools.

One observer in the field has noted that if this were 1485, we would have the choice of investing in improving quill pens or in the printing press. Similarly today, developing the new technologies can be the means to move us into a new era of educational productivity.

Consider the following:

- Where would you expect to find a group of kids—gathered around a textbook on fundamental procedures for flying or around a graphic display where one of the kids is bringing a 747 in for a landing at Los Angeles International Airport?
- How would you rather learn to program in Pascal: by reading a text on syntax or by writing programs suggested to you by a computer tutor that has

demonstrated the techniques you will need, and which critiques your programs?

In short, what sets one's imagination working on how to use the most important breakthrough in education: intelligent 2D simulation.

Why simulation?

Because it has a number of potential advantages:

- Productivity can be high. A simulation permits irrelevant factors to be carefully controlled or eliminated, so the user can often receive a far higher density of experience with the simulation than is possible in the real world. For example, pilots practicing landings using a simulator can repeatedly begin their practice at the final approach, whereas pilots in the actual aircraft must repeatedly take off, then maneuver in to the final approach. This takes time.
- Simulations cost less than using the actual equipment or constructing the actual situation. Fuel savings is a major advantage in the landing example above.
- Simulations can be fun and motivating for students if it is carefully designed and tried out.
- Using a simulation, dangerous experiments or maneuvers can be tried which would be foolhardy in the real world.
- Simulations can compress or expand time, so that the user can grasp events which might require years to observe in the real world, or which might take so little time as to be out of the realm of human observation.
- Simulation limits what can be termed the lexical loop:

(a) transfer a hands-on procedure to a verbal representation.
(b) have the student learn the verbal representation.
(c) have the student transfer from the verbal back to actual job performance.

Two-dimensional simulations are representations of dynamic systems that can be presented on a computer terminal or TV monitor. Two-dimensional simulations use pictures and graphics instead of real equipment like switches and meters. One of the major advantages of 2D simulation is that one set of trainer hardware can provide interactions with a wide variety of events and processes to be modeled. For example, the same TV screen and computer can present a medical diagnosis and a simulation of the Battle of Gettysburg in the Civil War. This flexibility is not possible with 3D simulations, which resemble a specific piece of equipment.

Types of simulation

Table 1 presents the major simulation features. These are simply suggestive of some design alternatives. The features are for the most part self-explanatory, though some elaboration is in order for the first five (IA Oct 79).

Applications: Operating procedure, diagnosis, and analysis of systems are the most frequent instructional 2D simulations. For example, two-dimensional operating procedure simulations are used by both American Airlines Flight Academy in Dallas and United Airlines Flight Operations Training Center in Denver. These flight crew training simulations use large Plato computer systems. Hazeltine in McLean, VA has also developed training for flight crews on aircraft systems operations. Hazeltine uses a minicomputer while Appli-Mation in Orlando, FL has developed similar training using micros. Much of such training will soon be on micros.

Outside of flight training, Wicat in Orem, UT has developed a simulation for army forward observer skills. Also the Plato Control Data Education Co. in Houston is developing an operator simulation for a specific chemical unit.

In the diagnosis area, Wicat has developed electronic maintenance training for automobile mechanics

A. Application	Examples of Applications
1. Operating Procedures	Pilots in an aircraft
2. Diagnosis	Medical diagnosis, electronic troubleshooting
3. Analysis of Systems	Modeling a complex chemical plant
4. Social/Team	Group problem solving
5. Personality	Simulation of a paranoid patient
6. Resource Allocation	Determining how long to spend with each customer waiting in a queue

B. Time Dependency
1. Time Dependent: Events occur based on time
2. Time Independent: Events occur based only on user actions and internal states

C. External Representation
1. Replica of appearance of system being simulated
2. Functional representation of major components

D. Internal Representation
1. Table driven (if a given condition is met, the specified action is taken)
2. Modeled by equations

E. Intelligence
1. Expert model
2. Model of student
3. Tutor

F. Purpose
1. Description/prediction of system operation
2. Instruction

G. Relation to systems simulated
1. Generic to several systems
2. Specific to one system

H. Interactivity
1. User operated
2. Feedback from computer or another person

I. Recordkeeping
1. Replay
2. Major events/costs

J. Input Mode
1. Keyboard/keypad with function keys
2. Controls (e.g., joystick, throttle)
3. Light/sonic pen
4. Touch surface
5. Voice

K. Output mode
1. Dials, gauges, lights
2. Electrical test points
3. Computer-generated images
4. TV scan of mockup
5. Stored images (e.g., microfiche, videodisc, slides)
6. Recorded motion (e.g., videodisc)
7. Voice

Table 1. Major simulation features

on the repair of electronic ignition systems using the Plato system. My own work at Wicat includes developing a 2D simulation for troubleshooting a complex radar system. The simulation uses the company's Distributed Instruction System which employs the new videodisc and MC68000 microprocessor technology.

The behavioral technology laboratories at the University of Southern California has developed and tried out an extensive micro-based electronic equipment maintenance trainer (EEMT). Using funds from the Naval Personnel Research and Development Center in San Diego, EEMT expects to get widespread use.

In medical diagnosis, Wicat has developed an extensive simulation for gastroenterologists. This work for Smith, Kline, French pharmaceuticals uses a computer-controlled videodisc, and has received international recognition in the medical community.

A number of universities are developing simulations to help students analyze complex systems in physics. Perhaps the best known is Al Bork's physics computer development project at the University of California, Irvine. This group has extensive experience developing materials for students, and has widely published what they have learned.

The University of Utah department of physics is developing a similar project using computer-controlled videodiscs. For years the Huntington science materials for high school students were developed by Ludwig Braun's group at the State University of New York at Stoneybrook. These pacesetting materials used the simple equipment available at the time, but demonstrated how effective simulations can be constructed.

Time Dependency: Simulations may be classified as time dependent or time independent. In a time-dependent simulation, events occur as a result of time (though that time may be lengthened or shortened for convenience). A time-independent simulation evolves as a result of user actions or changes in internal states that are independent of time. For example, in a medical simulation where a patient's critical condition would lead to death if the doctor does not take action quickly, the action may be considered time dependent. A simulation of radio system electronics, on the other hand, would likely be time independent.

External Representation: The external representation of a simulation refers to the manner in which it appears like the actual situation or equipment. A replica simulation appears, on the surface, as much as possible like the real thing. On the other hand, a functional simulation stresses underlying concepts. For example, in teaching fuel system operation to automobile mechanics, a complete layout of real tubing, pumps, carburetor, and the like could be used for a replica simulation. However, very simple graphics could be used to illustrate fuel flow and system operation, as well as permit the student to respond to faults. The replica simulation is more appropriate for teaching recognition of components and the physical layout of the system, while the functional simulation is especially useful for teaching system and troubleshooting logic.

Internal Representation: This may be simulated in terms of tables of actions or interms of equations. In the table driven approach, one entry in a table might be matched if a switch is in position 2 on an electronic panel. If it is, the simulation then shows that a meter reads 30. The entire system is represented as a finite number of states. The simulation program recognizes the current state, then travels down the table until it finds an entry which matches the current state. When the entry is found, the corresponding action is executed, resulting in a new state. In contrast to the table driven approach, a system may be fully modeled by a set of equations, where the number of possible states is so large as to be unmanageable in a table.

Intelligence: In this case, it refers to the ability of a system to model the student's state of knowledge about content, then tutor the student as needed. It does not necessarily imply the ability to converse with the student in English.

Game shows how it works

As an illustration of the components of an intelligent simulation, consider a game in which the user tries to locate a beast in a series of caves before the beast eats him. This popular game has a number of variants (e.g., Minotaur). The user cannot see all of the caves, and must construct a map of the dangers and possibilities in each cave. The system to be simulated is the caves, beasts, and their movements within the caves.

Given the information presented at any given time, certain inferences can be made about the location of the various dangers and opportunities in the system of caves. An expert model or procedure might make full use of this information in the most efficient manner possible. The student's actions might then be compared to this optimal solution, and inferences made about the student's strategies. This would become a model of the student.

Finally, a tutor function might compare the student's state of knowledge to the expert model to determine the bugs in the student's understanding of how to solve the problem. The tutor could then provide the student with problems in the caves which made those bugs especially manifest so that the student could correct the bugs (and miss the beasts).

Providing a tutor that can analyze the student's errors, find the areas of weakness, then prescribe appropriate study and practice can be expensive.

It is a matter of current research at Wicat to determine for what case it is important to know the student's bugs, and where it is sufficient to provide correction on the spot. Models of the student and provision for the tutor are characteristic of intelligent computer-assisted instruction (ICAI). The distinction between intelligent computer-assisted instruction and intelligent 2D simulation is thin. In both cases the student is given a rich environment in which to practice, along with tutorial guidance and modeling. ICAI tends to be more text oriented, while intelligent 2D simulation tends to graphics.

A number of researchers and centers are developing teaching strategies important to 2D simulation. Among these are Alan Collins at Bolt, Beranek, and Newman; Seymore Papert at Massachusetts Institute of Technology; John Seely Brown, Alan Kay and Adele Goldberg at Xerox Palo Alto Research Center in CA; and Avron Barr at Stanford. They have developed major programs over a number of years which are likely to form the foundation for intelligent 2D simulation.

The fruits of this work are beginning to appear in military systems: the advanced systems department at

Logicon in San Diego is developing a simulator to train ground controlled approach personnel. This simulator converses with the student in the very limited domain of aircraft landings. In addition Perceptronics in Woodland Hills, CA is developing an adaptive computerized training system (ACTS) for electronic troubleshooting. ACTS tracks a students's diagnostic and decision performances, and compares them to the expert. In the future, the lessons learned in projects like these will be applied in the classroom.

Military and commercial simulation

The armed forces have begun to invest in 2D simulation because it is relatively inexpensive and because the hardware can be used for a variety of equipment and procedures. The Army is now sponsoring AMTESS (army maintenance training and evaluation simulation system), which includes extensive 2D and 3D simulations for different systems. The Navy is sponsoring EEMT (electronic equipment maintenance trainer), which is another 2D and 3D system. The Air Force is sponsoring the FSTT (flight simulator troubleshooting trainer). These systems are not primarily research oriented, but are intended as precursors for larger development of 2D simulation in the armed services. They are all in the million dollar class.

A number of companies are now manufacturing 2D trainers. One form is a board simulator in which a large, fixed graphic representation of the equipment is displayed. A helicopter graphic may be shown on a large board, with blowups of critical electrical, fuel, and hydraulic areas where the student can make tests and obtain readings. Educational Computer Corp. in Orlando, FL, and Burtek in Tulsa, OK have both developed this type of trainer. In addition, Hazeltine, Plato Control Data, and Appli-Mation have 2D trainers on the market.

Computer hobbyists who develop excellent two-dimensional simulations are likely to find a market for their skills in the years ahead. Because simulations can be so engaging and yet relatively low in cost, they are likely to become widespread. Microcomputers offer plenty of computing capacity to construct simple two-dimensional simulations. Moreover, as 16- and 32-bit micros become available, some of the hard lessons we learned over the last decade about ICAI instruction can be applied to reasonable-priced intelligent 2D simulation programs.

Videodisc: This provides a new dimension at reasonable cost for 2D simulation. High-quality video graphics can be stored as individual frames on the videodisc, and up to 30 minutes of motion or 54,000 single frames may be selected in the optical reflective disc format.

Table 2 presents a number of groups which are developing computer-controlled videodiscs. Their work usually involves 2D simulation, primarily in the diagnosis of medical and electronic system problems.

Videodisc offers the capability to display motion, color, and audio. When these capabilities are combined with powerful computer software for the simulation, the result is a system that combines the advantages of the book, slide/tape, computer-assisted instruction, and videotape. The videodisc offers an advantage over videotape in that individual frames of text or graphics can be quickly and accurately called up one at a time, or as motion.

Group	Location
American Medical Association (Leo Leveridge)	Chicago, IL
Hughes Aerospace	Los Angeles, CA
Hughes Ground Systems	Fullerton, CA
Interactive Television	Arlington, VA
ISD	San Diego, CA
National Library of Medicine, Lister Hill Center	Washington, D.C.
Nebraska Videodisc Design/Production Group	U. of Nebraska, Lincoln, NE
Perceptronics	Woodland Hills, CA
Sandy Corporation	Southfield, MI
U. of Iowa (Joan Sustik)	Iowa City, IA
U. of Utah Physics Dept.	Salt Lake City, UT
Utah State (Michael DeBloois)	Logan, UT
Wicat	Orem, UT

Table 2. Some groups which are developing computer-controlled videodiscs

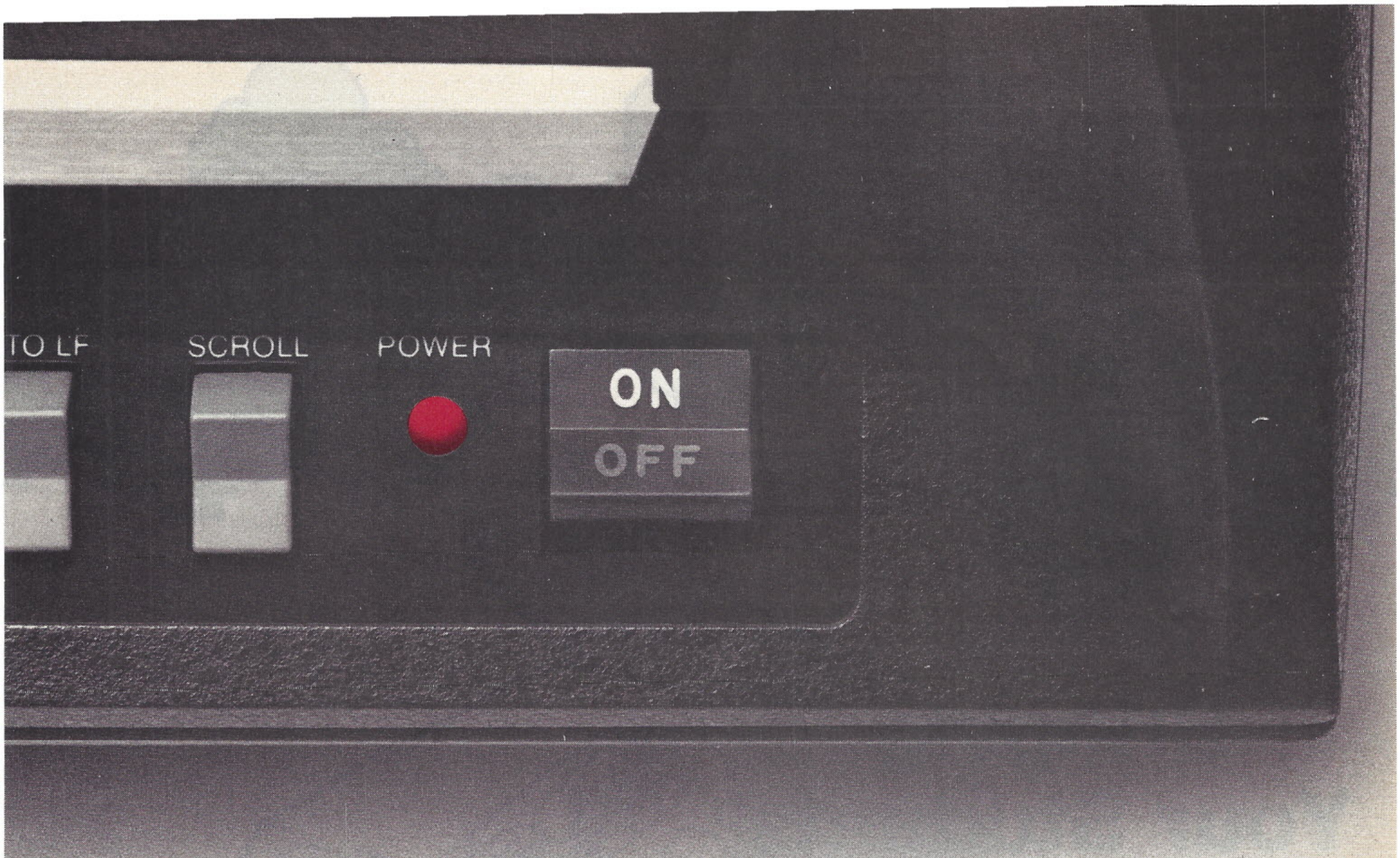
Job Aids: In many cases it is more efficient to provide job aids to a worker than formal school training for a skill which will be quickly forgotten without practice. A natural extension of intelligent 2D simulation systems will be the development of intelligent job aids. Some of the same processing intelligence which can guide the worker in diagnosing problems on the job. The tutor can keep track of faults as they occur and can construct a probability matrix which associates each symptom with a set of likely faults. In this way, the tutor can discover the most likely failures of a particular piece of equipment, and optimize troubleshooting for that particular piece of equipment by looking first for the most likely fault, given a particular symptom.

Simulation Authoring Systems: Although the hardware for a 2D simulation system is relatively inexpensive, development of programs remains difficult and, therefore, relatively expensive. EEC, Appli-Mation, and Wicat are developing or enhancing simulation authoring systems. At present these systems primarily model the system which is to be simulated, and, in the case of Appli-Mation, represent the correct troubleshooting procedure in a model of the expert.

It is not unreasonable to expect that simulation authoring systems will be extended to modeling student states of knowledge and providing a tutor function. These functions, when they are used at all, are now often coded using general purpose high-order languages. As we begin to better understand the structure of a tutor and the aspects of student bugs in a particular domain, we will be in a position to construct a simulation authoring language for these areas as well.

Intelligent 2D simulation is a way to aid and replicate the work of good teachers. It stresses highly interactive graphics-oriented displays. By inferring what the student does and does not know about the content, then adapting what is presented, the system can simulate a human tutor as well as the subject matter. This capability can be a breakthrough in learner productivity.

If this were 1485, we would have the choice of refining quill pens or developing the printing press. Books have now become important tools for teachers. There is a reason to believe that 2D simulation can similarly extend the teacher's capabilities. What is needed now are creative designers and pragmatic experimentalists to open this new frontier of human achievement. □



Diablo printers spend more time in this position.

A printer isn't much good if it can't do the job when it's needed.

That's why, at Diablo, we don't just design printers that work. We design printers that keep on working. In fact, we make them so reliable, you can just open the carton, plug in and play.

Diablo offers the widest range of reliable printers and options to give the flexibility you need. Which stands to reason. After all, we pioneered the daisy wheel technology and we're still the leader in it.

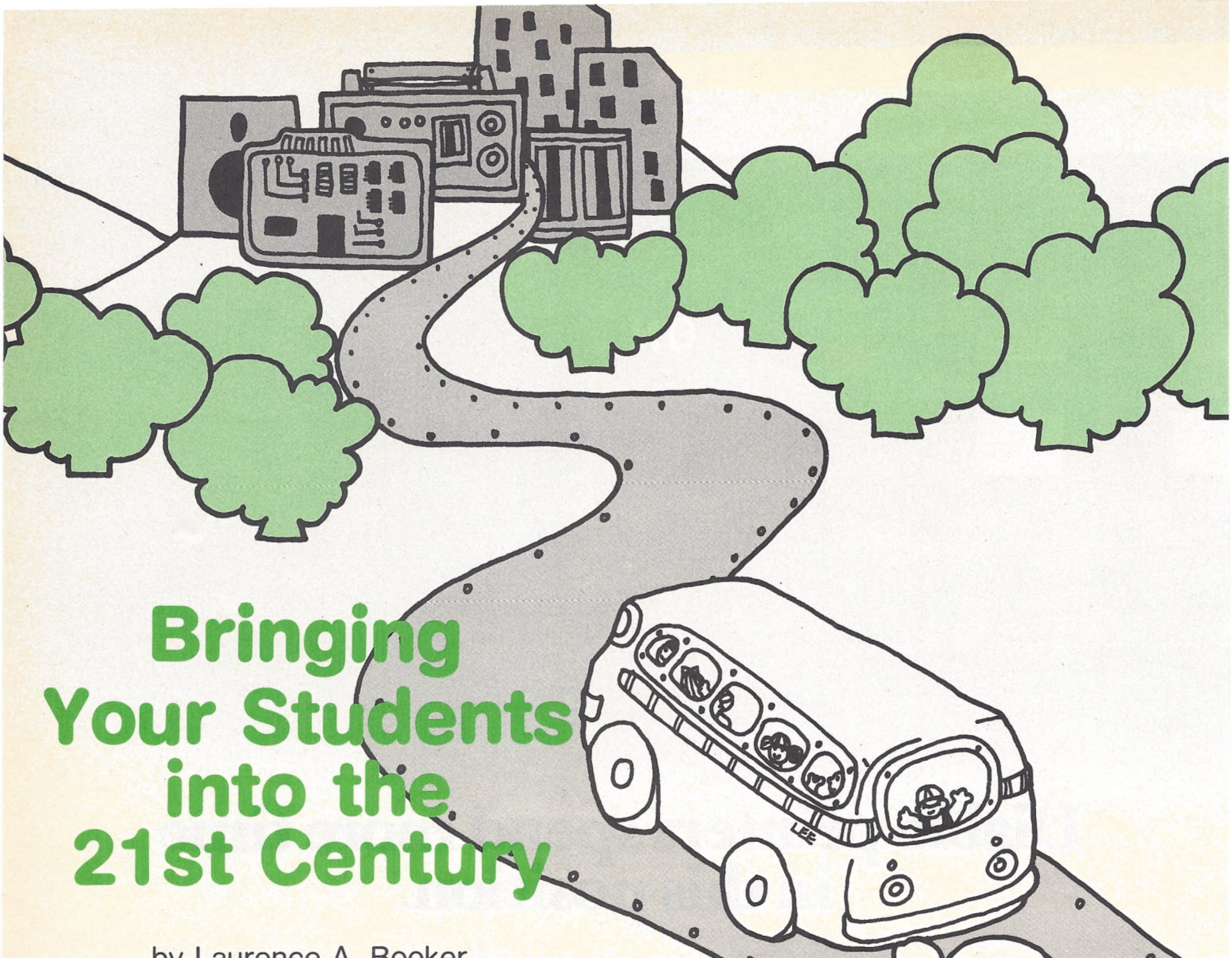
So if your printers spend too much time in the "off" position, you know what to do.

Switch.

Diablo Systems

XEROX

CIRCLE INQUIRY NO. 8



Bringing Your Students into the 21st Century

by Laurence A. Booker

During a recent luncheon with a group of computer teachers, the question arose of computer literacy among high school students and graduates. Most students, it was agreed, pass their high school and/or vocational school years up to their ears in math, English or trade and industry courses without a minute's exposure to data processing. Except for a few who took one or two courses in Basic, few graduates ever see a computer. They have no concept of terminal entry, a CPU, DASD devices, timesharing operations, corporate data processing, the computer's effect on society or how they will modify the future. Plainly, computer education just isn't here.

But it's beginning. Two years ago when the Addison County Vocational Center started its computer concepts course, the staff had to bring 11 years of industry data processing experience into the classroom. It had little idea of where the course was going, and only knew the bumpy route to be taken in trying to educate secondary school kids about computers. But in the brief time the course has been running, the initial 9-week course has blossomed into a 2-section double period class. The center's off and running to instill computer literacy among its students, and prepare them for the computer jobs that abound in manufacturing, distribution and all business offices.

The bulk of enrollees were business students—bookkeepers, accountants, secretaries. But also a smattering of college prep kids have enrolled. The course had to be redesigned to fit three basic student categories: the vocational student (business or

trade and industry, interested in developing marketable job skills); the academic student going on to college in math, computer science or a related field; and the inquisitive student who wants to learn about computers but may be undecided on a career. All three are welcome, not only because of the educational commitment, but also because it makes explaining an accounts receivable balance forward situation to a math student who only speaks trig more interesting.

The hard part is teaching them to speak the language. The classroom used by the staff has about 300 blank punch cards taped to the walls. On each, in heavy, black magic marker, is a computer-oriented phrase like DASD, microsecond, tape reel, packed numeric data, etc. They virtually cover the entire wall and teachers say they can hear the gasps from students when they first see this formidable array.

Learning—off the wall

"But don't worry," they're told. "Read the walls, learn the terms as you go along, and you'll be speaking the language in no time." Of course, the kids don't believe it. But this saturation technique works. They are bombarded by the words on the walls which are then related to concepts and ideas. Students are encouraged and gently prodded into using technical terms during class discussion. Phrases such as "it thinks" or "put it into the computer" are not allowed. Instead, "execution time," "disk file access," and "calculation" start to crop up. The old "illiteracy" begins to fade. Vocabulary takes on significance and meaning. The computer words from the walls become part of their language. They start to speak computerese. Saturation learning, doled out, moderate but explicit, becomes a highly successful learning tool. Disk access and linkage editor become daily terms, and computer confidence builds. There's also a gratifying snowball effect.

The computer concepts course is divided into two semesters. During the first, terms and phrases on the classroom walls are explained, and the concepts presented in the books and manuals are discussed. During the second semester, vocational students do some programming and terminal and computer operations to sharpen job skills. College-bound students get plenty of programming and systems work. And the general student may do a bit of programming, some operations—both terminal and computer—and work on a series of research papers.

The crux of the course this first semester is concentrated emphasis on computer lingo. Experience has shown that a student who concentrates only in programming, without the knowledge of an operating system, control language, or language translator gets bogged down and frustrated when "things" don't work right.

Part of the course shows how to use a technical manual. This forces a student to ferret out pertinent source material, important utility programs, control language options, advanced programming techniques like self-indexed arrays, advanced topics like logical and physical channels in a distributed data processing network plus a myriad of other items relevant to the course.

The teachers attempt to put subject matter into pespective during a class lecture first period Monday through Thursday. Preprogrammed practice set operations, simulating an inventory/order entry situation, an advertising/marketing problem and stock exchange trading are performed on the computer terminals duing the second, or lab periods. Some of the more advanced or ambitious students may start writing simple Basic programs during the first semester. About a week is spent on an introduction to Basic to show the class what a program is like and how it functions.

Friday's first period is reserved for a test of the week's work. The "festivities," as they are called, are comprehensive so that, if the teacher is spending a week on teleprocessing, the Friday festivities may include questions from previous weeks on ISAM processing, disk and tape structure, program language environment or data hierarchy. The teaching staff has discovered that this makes kids work a bit harder, and continually reviews the material.

A major topic is introduced each week and each has its relevant vocabulary. Hence, reading the walls and assigned text work during a particular week, highlights the terms for that topic and builds student literacy.

Beginning with the five major parts of a computer system—hardware, system software or operating system, programs, control language and data—the first semester builds each week to a new and progressively more challenging subject. The building blocks during the initial weeks are the fundamentals upon which more advanced subjects depend.

Course curricula

The second through the sixth weeks are spent examining the hardware itself: input, output, auxiliary storage, CPU and system console. Each part is broken down into its components and examined during a unit week. For example, data, as a major unit, is examined in its alphabetic, numeric and special character form.

Data is defined as any kind of information. This information is translated into punch card code used on a standard 5081. Punch configurations, comprised of digits and zones, are drawn on the board so that students can see how basic data is captured. Although the punch card is demonstrated, the point is stressed that punch cards are becoming a recording medium of the past. Diskettes for recording data are now in the forefront, and it is this kind of technological change that the teachers give the students.

The operating system—system software—is explained in terms of monitor, linkage editor, language translators, IOCS, sorts and other modules. The salient point is that operations going on inside the computer are all logical, directed executions, and not random electrical charges flitting to-and-fro amid the iron and glass.

Programs and their environments—business or scientific/engineering—are scrutinized. One week compares programming languages and their purpose—Fortran versus Cobol—and their four categories—file description, input, calculations, and output—as to whether these categories are explicit as in RPG II and Cobol or implicit as in Basic and Fortran. The point is that a program parallels a computer system in input, calculations and output.

The control language and its role in the system are defined, in addition to its unique powers.

After ideas have been thoroughly digested, the course proceeds to file structure, data hierarchy, stored program concepts and more advanced topics

such as teleprocessing, distributed data processing, timesharing and an introduction to virtual storage. Again, each subject covers about a week. Without basic building blocks, students would have a hard time learning the more difficult material.

On top of classwork and readings, each student is required to give an oral presentation on a computer-related subject. This requires individual research in a technical library, and each student becomes somewhat of an expert on a particular topic. In addition to expertise, he gains in developing tradecraft and enlarging his use of technical manuals and computer source material.

Teachers have their say

The section of the course that has become quite rewarding is corporate structure. Originally conceived for one week, it evolved into a 2-week learning experience composed of teacher lectures on the computer's place in the corporation, and the three main divisions of a typical data processing department. Whether the example is a large data processing department in a major corporation or simply a one-man shop in a local business, it still has three divisions: data entry, operations and programming, and systems.

These divisions are covered in detail with special stress on downtime, third shift, prime time, backup, management information systems, systems analyst, shift operator, supervisor, multiprogramming, multiprocessing, timesharing, etc.

The second week attempts to bring this corporate structure alive by a "systems simulation." A corporation is created and each student gets a job. A bookkeeping student becomes vice president of finance, a college prep guy becomes vice president of research and development and chief scientist, while another is director of marketing. All class members get a job in the corporation and all the jobs must be filled.

This past year the students in the computer concepts class chose to be a car stereo distributing company named Sound Sensation Inc., with annual revenues of several hundred million. Corporate headquarters were in Honolulu, and components were imported from Wales, Great Britain, West Germany and the Far East.

Sound Sensation Inc. had corporate board meetings and, under teacher prodding, became—if not fiery—then spirited. Company directors and vice presidents argued out corporate problems. . .usually financial. The biggest decision to come out of the meetings was if and when to go from a teleprocessing network to a distributed data processing network.

On-the-job problem-solving

The purpose of this, of course, is to parallel real-life. Industry is definitely moving in the distributed network direction. This was just the right sort of problem for a classroom exercise. Everything had to be cost justified and all members advised on software and hardware configurations.

This week is worth the entire course, because it brings the realm of data processing alive and forces students to employ all the concepts they studied during the semester. They argued baud rate, backup procedures, network configuration and operating system requirements.

Although the haggling and arguing are financial, the bulk of the work is data-processing oriented. And here's where literacy pays off. The "chief scientist" needs more, faster data collection capability from his car stereo lab in Cincinnati. He requires a quick-response real-time network. The manager of data processing operations, usually a vocational student developing operations skills, needs to worry about polling his remote minicomputers for sales data collection. Should he budget for a larger host mainframe? If so, how big and how much? A bookkeeper and vice president of finance say no, we can't afford it. What about the director of marketing? How can she forecast consumer trends without current sales, marketing data and analysis?

These students are using their computer literacy. . . and in a situation that prepares them for wage-earner roles.

One student asked a teacher why it was necessary to learn all this complex stuff rather than simply learning Basic and taking one of the many programming jobs advertised. The answer was easy. These are the routes the computer industry is taking: distributed data processing, satellite transmission, applications programs and database management.

Programming staffs will probably go the way of the punch card. There will always be a need for them, especially in software development; however, many companies are easing into the database environment and not replacing programmers lost through attrition. Databases provide tons of information for the clerical and administrative staff with desk-top video terminals. The vocational student who understands terminal usage and can communicate intelligently with a database manager is a student with a job.

Firms get the message

Computer literacy is essential for the business student entering the job market. Companies with no computer or terminal capability now, will, to stay competitive, have access to databases that perform timely financial reporting. Vocational students—bookkeepers and accountants—need to prepare early for financial computer-oriented work. Even the college prep student, who now finds himself in vocational computer courses picking up subjects other than straight math, needs to know about computers, their jargon, use and structure. Programming is only a small part of the entire data processing picture. He who denies these facts is woefully misinformed.

And where does this lead us? To the simple fact that to produce tomorrow's workers—let alone leaders—computer literacy must be promoted in secondary classrooms. While there may be some interesting and innovative technique used at the center, they by no means constitute the only comprehensive course of study. Constant curriculum changes are necessary to introduce, or at least roughly explain, the latest computer advances.

The overriding responsibility of educators is to expose youngsters to computer literacy. It's time to react to the question of "Why can't Johnny compute?" Johnny must be able to compute—as an operator, a programmer or a data entry person because the jobs of today demand it. □

A Modem Interface for the 80s

(Z-80s and 8080s That Is)

by Neil M. Harrington

If you are thinking about a modem for your personal computer but haven't settled on the best approach, here is one to consider: a software driven acoustic coupler. This is applicable to just about any computer, even one without a serial I/O port.

The software will show you how to: interface an acoustic coupler through a parallel I/O port, produce any desired baud rate, and make your personal computer act like a remote terminal.

A simple hardware interface for connecting an acoustic coupler to your computer is one bit in and one bit out from a parallel I/O port. Of course, if you have a serial I/O port, it will considerably simplify the interface.

The complete modem interface program is presented in listing 1. It is written in Z-80 assembly language using Zilog mnemonics. However, only 8080-compatible instructions are used. A quick scan shows that the program is highly modular. This costs a few extra bytes of instructions, but should make it easier to adapt the program to your system.

A flow diagram showing the main program functions is presented in figure 1. Upon entry, the program initializes the video display, then drops into the main routine at lines 400-590 in listing 1. The main routine is an endless loop that alternately polls the modem and keyboard looking for input. If modem input is detected, subroutine 'rmodem' is called to fetch the character; 'dchar' is called to display it. If valid keyboard input is detected, the character is sent to the modem via subroutine 'wmodem.' This process continues until ESC is pressed to exit to the monitor.

Scheme of actual data byte

The modem I/O drivers transfer data between the modem and the computer, translating between serial and parallel data forms in the process. Before examining the software techniques for doing this, let's look at the serial representation of a typical data byte. Figure 2 shows the serial transmission of E which is 1000101 in seven bit Ascii code.

Reading figure 2 from left to right, we first see that a logical one is transmitted between characters. The beginning of the character is signalled by a logical zero start bit, followed by the seven Ascii data bits in reverse order and then a parity bit. Finally, the character is terminated by one or two stop bits at logical one.

The parity bit is provided as an optional data check. It is set either to one or to zero by the transmitter to

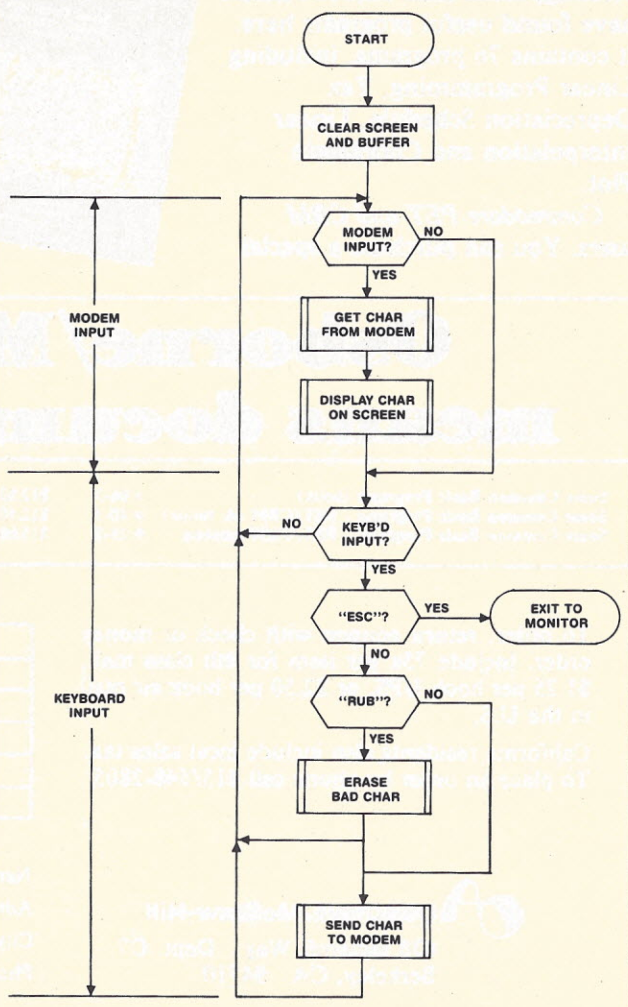


Figure 1. Program flow diagram.

make the sum of the eight data bits an odd number (for odd parity) or an even number (for even parity). The receiver checks the sum of the eight bits to determine if a transmission has occurred. The parity bit is usually ignored by time-shared computers, so it is not implemented in the modem drivers. However, only a few additional instructions are required to include parity if desired.

Now let's look at the modem driver software. Subroutine 'rmodem' (lines 820-940 in listing 1) accepts serial bits from the modem and returns with the input character in the accumulator. 'Rmodem' is called as soon as the start bit is detected (i.e., the signal drops from one to zero). 'Rmodem' calls 'delay1' at line 830 to wait for the center of the first data bit to arrive. Each of the eight data bits is read in turn from the least significant bit (LSB) of input port 'inmodm.' As each bit is read into the accumulator, it is added to the bits read previously and saved temporarily in the D register. The precise time between each bit is obtained by the call to 'delay' at line 900. On exit from 'rmodem,' the completed data byte is in the accumulator. At line 930, the most significant bit (MSB) is set for proper handling in the display routines.

An Ascii character input from the keyboard goes into the accumulator. From there, subroutine 'wmodem' (lines 1150-1310) transmits it to the modem a bit at a time from the LSB of output port 'otmodm.' First a zero start bit is sent (lines 1170-1190). The eight data bits are sent out one at a time starting with the LSB (lines 1200-1260). The precise time required for each bit is provided by the call to 'delay' at line 1240. Finally, two stop bits are transmitted at lines 1270-1300.

Software baud rate generator

The precise time between bits is provided by a short subroutine 'delay' (lines 1350-1410). This is the heart of the modem interface, so let's take a closer look. 'Delay' consists of two tightly nested loops. The number of times the loops are executed depends on the integer timing constants TIM1 and TIM2. Since the values for TIM1 and TIM2 will differ for each baud rate and CPU clock frequency, we will need a convenient way to compute them.

The time per bit produced by the timing loops can be expressed by the following equation:

TL = (C*(14*B + 21) + 73)/F where
C = TIM1
B = TIM2
F = CPU clock frequency

The constants 14, 21 and 73 were obtained by adding the number of CPU cycles required to execute the instructions in each loop. (The value 73 includes the average number of cycles for lines 850-920 of 'rmodem' and lines 1200-1260 of 'wmodem.) If F is entered in megahertz, TL will be in microseconds.

The required time per bit for a given baud rate R is given by TR = 1.E6/R, where the constant 1.E6 causes the value for TR to be in microseconds. Equating TL with TR and solving for B to the nearest integer, we obtain

B = INT(((1.E6*F/R−73)/C−21)/14 + 0.5).

The resultant percentage error in the time per bit is E = 100*(TL−TR)/TR. Substituting for TL and TR, we obtain

E = 100*((C*(14*B + 21) + 73)*1.E−6*R/F−1).

The optimum values for the timing constants B and C are those that produce minimum error, E. These values can best be found by systematically selecting values for C and then computing values for B and E from the equations above. It should be noted that C must be an even number because a third timing constant, TIM0, is C*3/2 to time a bit and a half.

To make it easy to determine timing constants, the whole procedure outlined above has been put into a Basic program. The program is given in listing 2, and a sample run for 300 baud at a clock frequency of 2.5 MHz is shown in the table, indicating that minimum timing error is achieved with C = 4 and B = 146 or with C = 20 and B = 28. The timing constants are entered in the assembly program at lines 2600-2620.

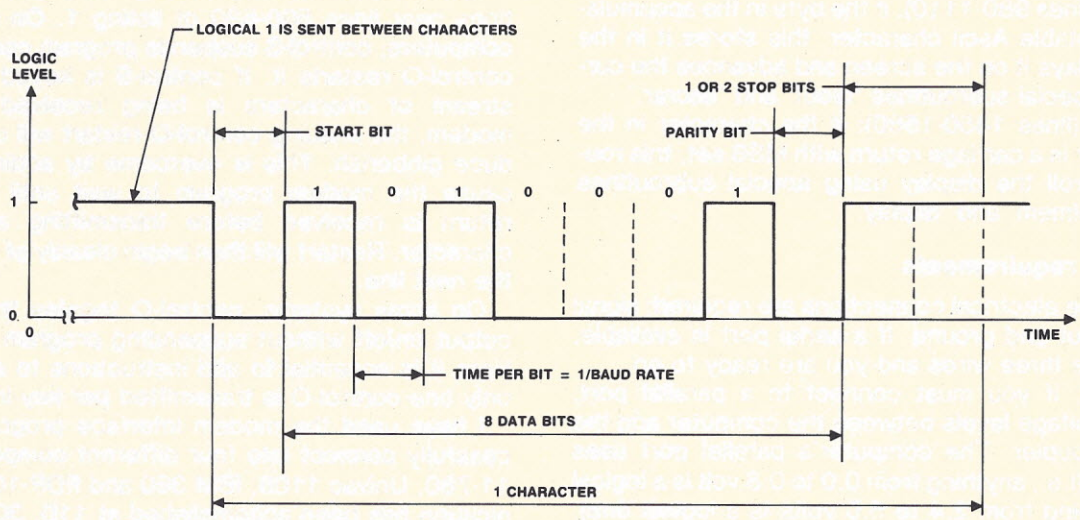


Figure 2. Transmission of Ascii character, E = 1000101. (Note: bits are sent in reverse order)

The I/O functions performed by the modem interface program are now described. Since computers differ widely in the way they interface with the keyboard and video display, it is unlikely the listed I/O routine can be used as is. Therefore it is important to understand how they work so you can modify them for your computer.

A personal example

The keyboard on my system works as follows: while a key is depressed, its Ascii code is present at input port 'inkeyb.' When the key is pressed, the MSB is strobed (i.e., goes briefly to logical one and then returns to logical zero). To avoid sending a character to the modem more than once per keystroke, the strobe bit is tested at line 540. If the strobe bit is off, the character is ignored unless it is a control character. The character displayed after a keystroke is the one echoed by the host computer to show that it was received properly.

The video monitor on my system displays 1024 characters as 16 lines of 64 characters each. Characters are sent to the screen via output port 'otscrn.' All characters appearing on the screen are contained in a 1024 byte circular buffer starting at address 'buffer' and ending at 'endbuf.' It is not necessary to start the buffer on a page boundary; it may reside in memory wherever you choose. Characters in the buffer are never moved. Instead, pointers to the current first and last lines (top and inline) are maintained. When it is time to scroll the display, these pointers are simply moved down to the next line in the buffer. If this action moves a pointer beyond the physical end of the buffer, the pointer is reset to the beginning.

Here are brief descriptions of the three main I/O routines:

RUBOUT (lines 630-780): used to erase an incorrect keyboard input. The bad character is removed from the screen and the buffer. A delete is sent to the modem, and the next three characters are intercepted to suppress their display. (These characters are "backslash-badchar-backslash".) Special routines 'erase,' 'bspace' and 'space' are used.

DCHAR (lines 980-1110): if the byte in the accumulator is a printable Ascii character, this stores it in the buffer, displays it on the screen and advances the cursor using special subroutines 'tvout' and 'dscrsr.'

LNFEED (lines 1450-1560): if the character in the accumulator is a carriage return with MSB set, this routine will scroll the display using special subroutines 'movelp,' 'filmem' and 'dsplay.'

Hardware requirements

Only three electrical connections are required: signal in, signal out and ground. If a serial port is available, hook up the three wires and you are ready to go.

However, if you must connect to a parallel port, translate voltage levels between the computer and the acoustic coupler. The computer's parallel port uses TTL levels (i.e., anything from 0.0 to 0.8 volt is a logical zero; anything from 2.4 to 5.0 volts is a logical one). The acoustic coupler uses RS-232C levels: +5 to +15 volts is a logical zero; and −5 to −15 volts is a logical one.

There are a number of simple circuits for effecting the required voltage conversions. I chose MC1488/1489 integrated circuits specifically designed for this purpose. The complete hardware interface is shown in figure 3.

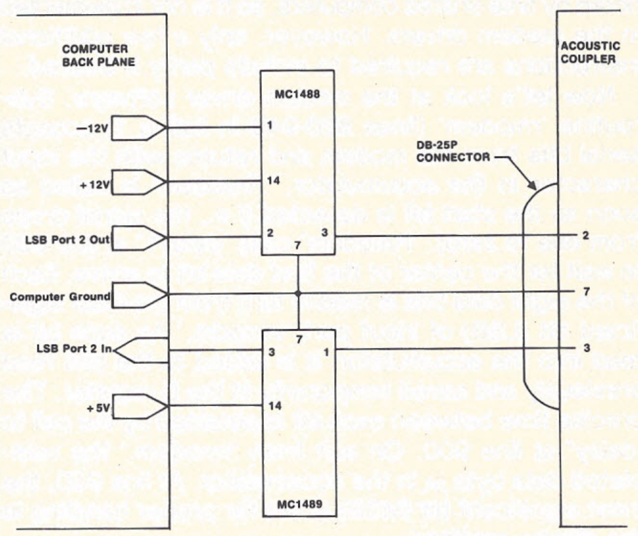


Figure 3. Hardware interface for parallel I/O port.

Here are brief summaries of some program changes that may be needed to customize the software:

Timing constants (lines 2600-2620 in listing 1) will probably have to be changed for your computer. The Basic program in listing 2 determines the best timing constant values for your computer's clock frequency. In the program output, C is TIM1 and B is TIM2.

Keyboard and video routines of the modem interface software will require the greatest amount of change. To design them effectively, it will be necessary to learn how I/O is handled in your system. One note of caution: if I/O functions are interrupt driven, disable interrupts on entry to subroutines 'rmodem' and 'wmodem,' and reenable them on exit. This is essential to avoid interruption during the timing loops.

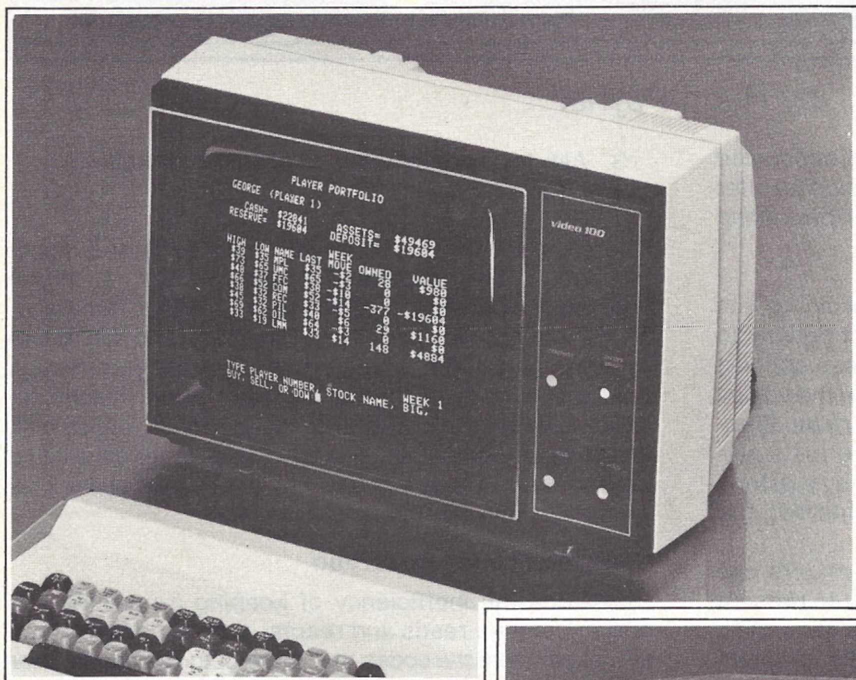
Special control characters can be added to instructions near lines 500-530 of listing 1. On some host computers, control-S suspends program operation and control-Q restarts it. If control-S is keyed in while a stream of characters is being received from the modem, the ensuing control-Q restart will usually produce gibberish. This is overcome by adding code to cause the modem program to wait until a carriage return is received before transmitting a control-S character. Restart will then begin cleanly at the start of the next line.

On some systems, control-O toggles the program output on/off without suspending program execution. Here it is essential to add instructions to ensure that only one control-O is transmitted per key-in.

I have used the modem interface program to successfully connect into four different computers (VAX/11-780, Univac 1108, IBM 360 and PDP-10). Communication has been accomplished at 110, 300 and 600 baud. It works! If you decide to implement a software modem interface, you will learn a lot about your personal computer. And you might even have fun doing it. □

Program on Page 134

PROFESSIONAL B/W MONITORS
Designed for industry...priced for the home.



video 100

The video 100 computer monitors are ideal for all your personal and business needs. These highly reliable 12" black and white monitors feature a 12 MHz band width and 80 character by 24 line display. Plug-in compatability with Apple, Atari, Radio Shack, O.S.I., Micro-Term and Exidy make these the perfect text display for almost any system.

Sturdy, lightweight plastic cabinet

UNDER $160.00

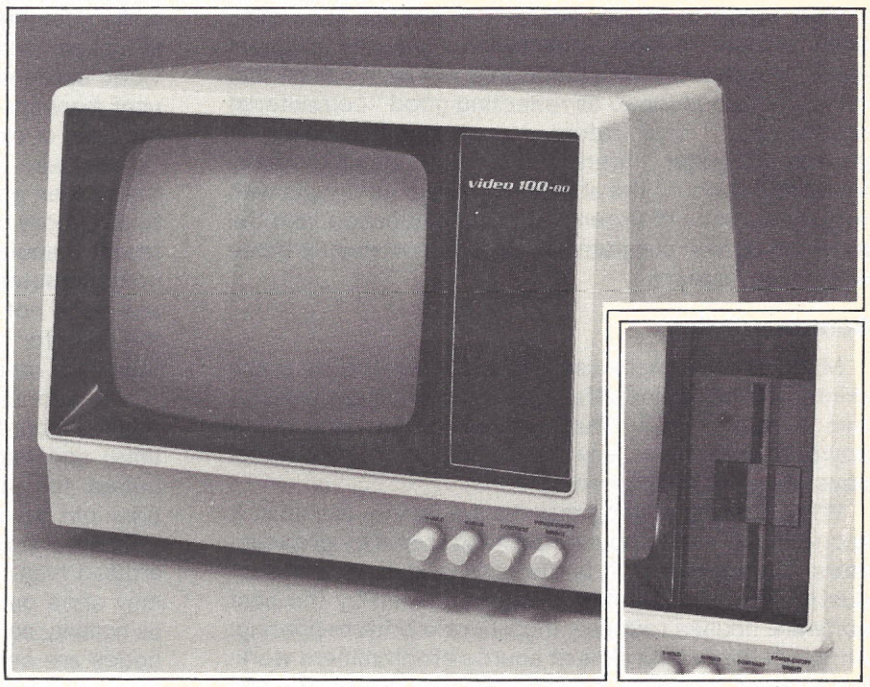
video 100-80

The model 80 features an industrial grade metal cabinet with built-in disk mounting capability and space for an 11" x 14" PC board for custom designed electronics.

The solid state circuitry assures a sharp, stable, and trouble-free picture. The front panel controls include power, contrast, horizontal hold, vertical hold, and brightness. Adjustments for size, video level, and width are located on the rear panel.

Rugged metal cabinet with disk space

UNDER $200.00



VIDEO 100 AND VIDEO 100-80 SPECIFICATIONS

- 12" diagonal measure display
- Convenient front panel controls
- Video bandwidth 12MHz ±3 DB
- Input impedance 75 Ohms
- 80 character by 24 line display
- 90° deflection picture tube
- Video 100-80 provides mounting space for mini floppy disk.
- Resolution—Over 700 lines at center horizontally—over 350 lines at center vertically

LEEDEX CORPORATION

2420 East Oakton Street, Suite E • Arlington Heights, Illinois 60005 • (312) 364-1180 • TLX: 25-4786

Dealer discount available

IMPROVING YOUR CONSOLE INPUT

by Hugh Poynor

This is the first of a 2-part article, to be concluded next month, on how to write programs in Basic that offer improved screen formatting, effective user prompting and convenient user input.

Sitting down at a computer video console to run most programs requires that the user select options to meet the unique requirements of his job. These options, commonly referred to as job parameters, are the values given to important variables used by the program. Parameters specify which of many operations are to be performed: the account to be displayed, the quarterly period to be summarized, the data files containing the records of interest and so forth.

What is needed is to show how programmers can fully utilize the ability of computer programs to present stored variables as job parameters and ask for a minimum amount of new information from the program user. To achieve this, procedures will be covered that can best be described as reflecting good "computerist psychology." This refers to an understanding of person-computer interactions that simultaneously takes into account the abilities of users, as people with varying degrees of urgency and sophistication, and the power of stored program computers, particularly those with video screens.

Screen printing

Many programmers use the video screen as a line printer, scrolling line after line of output from the bottom of the screen. They are using the computer's power in reverse, at least insofar as good computerist psychology is concerned.

While bottom-printing may be a primitive example, it is one of the most popular bad habits used by programmers. Unfortunately, it is rife among leaders in the industry. Popular operating systems written by the best software houses routinely incorporate bottom-printing. Although this article aims at source programmers working at home, hopefully it will find its way into the various software houses.

What follows are a few principles to take into account. The first concerns screen printing.

> *Use top-down printing that stops when the screen is full. Move only one line onto the screen at the bottom, thus giving a natural left to right and top to bottom page effect for easy reading.*

How can the top-down screen printing principle be programmed?

1. Clear the screen and home the cursor
2. Print only the next 23 lines (a 24 x 80 screen)
3. Ask for input in order to proceed or quit
4. Loop on (1)

The programming in this case operates to the disadvantage of the machine's efficiency. Several microprocessor operating systems (e.g., CP/M, Cromemco, MITS, Micropolis) have disk directory and file dump routines (e.g., 'dir,' 'type') that race the information past our eyes at a frightening clip. Often, therefore, users can only glimpse the top parts of larger disk directories. Sometimes smaller disk directories will fit a screen, and in those cases the directory display programs are fully useful.

Clearing the video screen

Besides the inefficiency of keeping a machine idle while someone reads and reacts, a more serious problem stems from the codes that manufacturers use to clear video screens. These are ordinarily documented in the user manuals, but sophisticated codes may be lacking in the case of older terminals with skimpy video controllers.

Generally, two operations are performed: clearing the screen and homing the cursor. This involves using two or more special codes which must be printed. Different manufacturers select different codes for some non-character printing simply because the Ascii standard does not establish the codes forthrightly. This is the case for homing and clearing the screen, for which there are no standards.

It is suprising that codes have been established by Ascii standards to position the cursor (one position at a time) and ring the bell, while the codes that are required for utilizing the fundamental screen printing principle have not been formalized.

The screen-manipulating codes are less well standardized than the Ascii character codes. Bottom-printing may arise out of this standardization problem, insofar as homing, screen clearing and other screen-manipulating codes are concerned. Realistically, there is small variance in the codes—possibly no more than three are employed by the video screen manufacturers. Therefore, the problem certainly does not stem from an overwhelming variety of codes.

Programmers can solve the unknown code problem in one of several ways: (1) determine the codes for the video screen on which the program will be operating, and then supply the program to the user appropriately coded; (2) use only the most popular codes, and of course, supply a caveat emptor for other screens; (3) use all three codes where only one is really required (this will not have undesired side effects, but it should be tested); (4) provide programs with a screen output subroutine arrangement wherein a user is required to

integrate his particular codes into source subroutine and then recompile and integrate the subroutine.

By these means, most programmers can employ the screen-manipulating codes that will permit application of console input principles for their programs. Alternative 4 has been coded in Basic for ADM, Hazeltine and Adds terminals.

The expected response to a program input prompt is often ambiguous because of unclear prompts or because of user experience with inconsistent alternatives allowed by the computer program.

Most Basics will hang up following an 'input' statement if (1) the entered material is not of the type (numeric or string) specified by the computer program; (2) if the number of separate entries does not correspond to the number expected. An entry of 'yes' when a numeric variable was specified by the program, or an entry of only two out of three required variables will invoke the Basic monitor and inform the user that some input value seemed to be illegal.

Good programs are written in a way that promotes user confidence

Good programs are written in a way that promote user confidence. They do not try imagination and patience during the interactions when job parameters are selected. The principle that corresponds to this aspect of person-computer interaction is: standardize user reponses.

Request input responses from users by asking explicit questions and when possible by using standardized terms and formats. Ask for natural-language (not code) responses and check them consistently.

Informing the user of optional responses to an input query such as 'okay to proceed (y/n)?' is an easy part of this practice. Other requirements for operating consistently with the principle are:

1. always use (y/n) or some other standard signal in the prompt;
2. never take a shortcut; and
3. always check the response to the same extent each time it is input.

Unfortunately, upper versus lower case and spelled versus abbreviated forms will be input unpredictably and must be anticipated by the programmer. Users that sometimes get away with non-standard responses will develop a mistrust of the program. The programmer should therefore be thorough and absolutely consistent in checking input. Subroutines are frequently the best way to handle these input check requirements, and to provide specific feedback to the user on detected input errors.

Before discussing some examples of user input checks, a word about user nervousness is timely. It is easy to destroy a file or an entire disk or tape by doing "the wrong thing." This involves user input blunders, such as updating on a new file rather than a history file, or backing up a disk with the drives switched, thereby destroying the original disk. After all, computers work as efficiently against us as they do for us. This principle tells us to be explicit in questioning users to help prevent these blunders. For example, where source and destination files are asked for the same input line, there should be a following statement and query such as, 'the disk on drive B will be copied onto the drive A disk destroying A's contents. Okay to proceed (y/n)?'

Alternative check method

Being explicit about tape or disk is useful to the extent that it relates to the tape or disk of interest and not to the drives. We copy the magnetic medium, not the drive with a copy program, so ask the question for which an answer is sought. These guideposts help users double-check their input and help to somewhat reduce user nervousness.

Checking numeric input can best be done by not requesting a numeric variable at all, but by supplying a string variable to the Basic input statement. This permits a tolerance check before the input is converted to a numeric quantity. Another point is the standard use of a certain precision in decimal places. You might always omit decimals or always include a certain number of places, just to make their presence a routine matter for users.

Break-out values are stopping points in a program and are always welcome when long lists are being input. They must be standardized and explicit rather than a matter of guesswork. Popular choices for these values are 'bye,' 'stop,' 'finish,' 'end,' 'subtotal.' Your choice should be clarified, of course, and unerringly used in all interactions. In the name of consistence, never allow a 'stop' to be input in place of 'bye' if the user has been informed that 'bye' is expected. Tolerance for lower versus upper case should be programmed, because text input can lead to an upper/lower pattern that starts each line in upper case.

Care must be taken that the break-out word is not part of the text, so require that the length of the input string be consistent with the length of the break-out word before checking for a word-wise match against the upper/lower case break-out options. Programming will be more efficient if the check itself is staged, such as BYE, Bye and bye, all having a string length of three. In cases when most input does match, but in some way the input is not acceptable, simply ask for clarification: 'was that bye (y/n)?'.

Finding input requests

In a 24 x 80 screen, there are over 1,900 possible places to hunt for input requests. Admittedly, only 24 lines are used for the input request except in rather exotic cases, but the standard screen principal further limits input to the uppermost two lines and moves status or option information to the screen area well below these lines. This lets the user anticipate where the input prompt will happen to appear and reduces clutter. A by-product of this principle is that, because a reserved screen area is set aside for options and status information, the programmer will tend to display these useful facts more often.

Each request for user input, if not a series of figures or lengthy text, should be displayed at the home loca-

tion in the uppermost left screen location. Generally, this is accomplished by homing the cursor and issuing an input statement with a prompt. At times, it may be useful to erase the screen before homing the cursor, producing a clean slate for gaining better user attention. Screens should not be completely erased, however, if the information might be advantageous to the user.

The second line can be used to advantage for checkout of the user response. For example, use this line after requesting a parameter in a copy job where reversed drives would be harmful. In these cases, use 'okay to proceed (y/n)?' or a more explicit statement line such as the one previously used for checking copy requests.

The present principle is one of the simplist console input guidelines and can be summarized as follows:

> *Displays in standard locations help focus the user's attention. Ask for input in the uppermost location of a clear screen. Additionally, display status information or options below the input request.*

This principle forms the foundation for the discussion of menu techniques below.

Distinguishing the program

Menus are employed to make a selection from among a set of items prepared by the programmer. This handy organization and display technique adds polish to the appearance of console display and leaves a good impression. When fully employed, menus permit control of programs and file selections as well as individual job parameters. In these cases, there are menu selections for further menus.

After selecting a particular program from a program library menu, it may be necessary to identify the file which is to provide data for the program. One item on the program library menu would then be data file menu. It is thoughtful to include a recursive feature so that program library menus have selections for data file menus and vice versa. In individual program menus that display the parameters for a particular job, a selection of the data file menu would be helpful if the user wishes to alter file specifications.

It is good practice to have values established for each menu item, so that the user will not have to enter all the job parameters each time the program is used. Give these to the user in the menu as default values. If he does nothing, the parameters suggested by the programmer will be used. In ways we will see shortly, it is very convenient for users to identify specific job parameters and alter them for the particular job at hand.

A sound technique is to allow the altered values to serve as defaults for the next usage, making the menu "learn" something each time. To do so will require storage of the defaults in a file that is read before each use. Give the file the same name as the program, but instead of a file type 'bas' or 'int,' make the file name 'men.' Having the same name as the program that uses it helps ensure that the menu file will be remembered when making copies or back-ups of the program and all its files.

Large menus are best organized into pages that fit attractively on the screen. The page should be titled either with the purpose of the program (e.g., File Sort Program) or with the theme represented by the group of items (e.g., Select Addressee for Routing Output). When several pages are required, one menu item must be devoted to 'backward page' selection and another devoted to 'forward page' selection on each page of the menu except, of course, the first and last pages, which can only allow selections forward and backward, respectively.

For visual satisfaction

Programming tips on centering titles and aligning items to be displayed are provided alongside examples at the end of this article.

Menu formats look polished when they occupy side-by-side locations, as in restaurant formats. This allows 40 characters for the parameter and its menu code value. A pleasing visual effect is attained when a centered title is placed above this double-column arrangement. Start the title at about the fourth line on the screen in order to allow several blank lines after the standardized input location. Print the two columns immediately below the title line.

Menu code values are a convenient shorthand for making selections. Choose code values that are numeric and standardized if possible. For example, an input prompt at the home location might ask 'choose the item you want to change (0 = end menu).' Whenever zero is entered, the program proceeds to the next task following the menu.

Once a user chooses an item code, erase the prompt and accept a new parameter as input for the item of interest. For example, after choosing an item code for altering the data file to be read, a prompt might ask 'enter file name.' Once it has been entered, the program reflects the name change in the menu, erases the last prompt and again asks 'choose the item you want to change.' This will continue until all the menu items have been altered to suit particular job requirements.

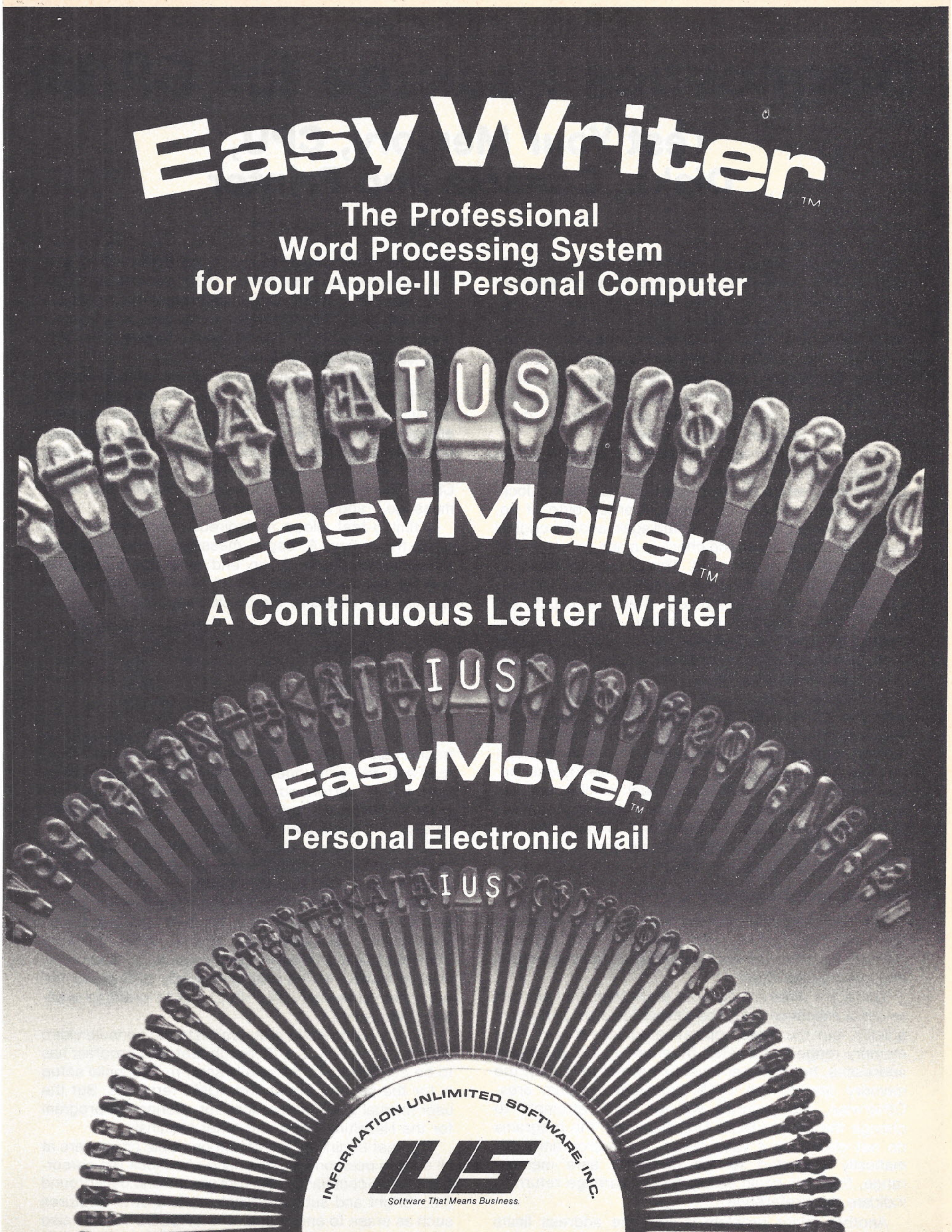
Flexibility is a premium

Notice that many items will not require further input once they have been selected to be changed. Whenever only two values are possible for a parameter (e.g., drive A or B, yes or no, ascending or descending), the computer program can simply exchange values without issuing a new prompt that requires further user input. In these cases simply make the exchange, reflect the new value in the displayed menu and again ask 'choose the item you want to change.'

Like top-printing, the menu procedures keep the computer waiting for a person to read and react. Our console input principles are machine-efficiency villains once again. The user, however, benefits. From the menu procedures discussed above, a new principle can be formulated.

> *Show the user as complete a picture as possible of the alternatives available as job parameters. Allow alternatives to be frequently and easily changed. Use menus for documentation of programs and data libraries. This allows "browsing" in the library.*

Next month Poynor will discuss inventory of programs, prompting techniques, value and appearance formats, along with programs.


EasyWriter™
The Professional
Word Processing System
for your Apple-II Personal Computer
IUS
EasyMailer™
A Continuous Letter Writer
IUS
EasyMover™
Personal Electronic Mail
IUS
INFORMATION UNLIMITED SOFTWARE, INC.
IUS
Software That Means Business.

Software Review

Diagnostics Package for CP/M:

Test Your Memory, CPU, Disks, Console and Printer

by Alan R. Miller

When microcomputing began with the Altair, unreliable dynamic memory boards were common. Memory refresh was performed asynchronously using a one-shot circuit for the timing. If refresh timing was not quite right, the data could spontaneously change. Basic took 15 minutes to load from a cassette tape (30 minutes from paper tape). But if the memory was acting up, Basic would malfunction and another 15 (or 30) minute loading sequence was necessary. A good memory test program became essential.

The simplest approach is to copy a memory location into the accumulator, complement it, and put it back. A comparison can be made to see that the memory location actually contains the complemented byte. If so, the accumulator is complemented again and the original byte is returned. A second check is made to insure that the original byte is restored. This simple approach to memory testing will discover grossly defective or nonexistent memory. Unfortunately, it will not usually uncover timing problems on dynamic memory boards. Neither will it find problems caused by shorted address lines. In this case, if one memory location is changed, other addresses will change too.

A fairly sophisticated memory test program is supplied by TDL with its 16K boards. Lifeboat Associates also provides a good memory-test program. But memory-test programs only check the main memory, not the CPU registers or instruction set execution. Neither do they test peripherals such as disk, console and printer. Realizing this, Supersoft has written a set of test programs called Diagnostics I. There is one version for all CP/M operating systems and another version for TRSDOS, each at $60.

Memory test programs

The two programs provided in the package perform the same tests. They are assembled for different memory locations, so each can test the memory region occupied by the other.

There are different memory tests that the user can select at execution time. The simpler tests are performed quickly, but the more sophisticated take longer. The memory range is indicated by entering 2-hexadecimal addresses, but the user must be careful to restrict the memory test to the CP/M transient program area. Otherwise, the program will die as it attempts to change the CP/M FDOS. Unfortunately, the programs do not check for a valid range. If the program automatically defaulted to the maximum safe memory range, the user could simply type a carriage return to indicate the preference.

Another minor problem is that the address limits must be entered as 4-hex characters. Thus, leading zeros must be given for addresses below 1000 hex. For example, one must enter '0100' rather than '100.'

A separate assembly-language program activates the CPU-chip test. The registers and all one-byte instructions are tested first. The program identifies the CPU type as 8080/8085 or Z-80.

If your 8080 is made by one of the second-source suppliers rather than by Intel, the CPU test may incorrectly report a bad CPU. The problem is that a design error in the original 8080 CPU was corrected by a second-source supplier. When the CPU diagnostic test doesn't find the expected error, it reports a faulty CPU. The manual explains how to disable this error message if you have a second-source 8080.

A timing test allows an indirect check on the CPU clock frequency. A 'begin' message is displayed and the console bell sounds; an 'end' message is displayed and the bell sounds again. The lapse time between the two can be related to the clock frequency:

Lapse time	Clock frequency
2 min	2 MHz
1 min	4 MHz
48 sec	5 MHz
40 sec	6 MHz

With the disk test, all regions except the system tracks can be checked. During this phase, the disk-read and disk-write operations are tested, and a track-seek test is performed. The CP/M version of the program is written in assembly language, but the TRSDOS version is a Basic source program.

The disk-testing phase demonstrates sensitivity. Lightly tap the computer table, or even the disk case itself, to see if a disk error occurs.

The line printer test program is initialized by giving the printer width and stating whether it is capable of lower-case and upper-case letters. After all Ascii characters are printed in each column, an Ascii form feed is issued. If the printer properly processes the form-feed character, this step should produce additional line feeds.

The final diagnostic program tests the console video screen. Since this is a thorough test, the program has to be configured to the actual screen. The initial setup is for the popular Soroc and ADM3 terminals. But the user manual gives instructions for altering the program for the Hazeltine 1500 and other terminals.

The test begins by displaying printable characters at all screen positions. Additional tests check the cursor-positioning commands, the foreground and background (i.e., bright and subdued displays) and other features such as erase to end of line. The TRSDOS version also tests the graphics capabilities of the video terminal. □

An Interrupt-Driven Keyboard Buffer

The typical microcomputer is directly coupled to the user's console through a pair of bidirectional serial or parallel ports. One port is termed the data port, the other is called the status port. The data port is used to transfer information from the keyboard into the CPU, and also to send data from the CPU to the console video screen or printer.

One bit of the status port is used to signal the computer that a character has been entered from the keyboard. The corresponding byte, of course, is located in the data port. Another bit of the status port indicates to the computer that a byte, previously sent to the console, has been displayed.

If the computer is directly coupled to the console, it must be slowed down by a factor of a thousand or more. A typical console-input routine might start with:

```
CONIN:  IN   CSTAT  ;CHECK STATUS
        ANI  INMSK  ;MASK UNWANTED BITS
        JZ   CONIN  ;LOOP UNTIL READY
```

The computer reads the status port and masks all but the character-ready bit with a logical 'and' operation. If there is no character to be read, the conditional jump will cause the computer to loop repeatedly through these three instructions. As soon as a character is entered from the terminal, the logical 'and' produces a non-zero result, and the computer executes the next part of the input routine:

```
        IN   CDATA  ;GET THE BYTE
        ANI  07FH   ;MASK PARITY BIT
        RET
```

The character is thus transferred from the keyboard to the computer. Because the computer is operating so fast, it can spend over 99% of its time looping around the first three instructions of the input routine. A more efficient way to transfer information to the computer is through a buffer area in memory. With this arrangement, characters are entered into the buffer whenever the user types them. This can occur even if the computer is busy with another task, such as printing information on the list device or the console. When the computer needs input, it is obtained from the buffer rather than directly from the keyboard.

This is the additional element needed to make the process work. The console keyboard is wired so that, when any key is pressed, the computer is interrupted from its current task. It then reads the input character, places it into the buffer, and returns to its prior task. When completed, the computer reads the buffer for the next task.

With a buffered keyboard, the user can type several command lines one after another. The computer will execute each command in turn. The number of commands that can be typed ahead in this way is limited only by the buffer size.

Adding and extra byte

The listing gives a complete user interface program for the Lifeboat version of CP/M. The sections pertaining to the interrupt version begin with a row of asterisks and end with a row of semicolons. A keyboard entry interrupts the processor causing a call to address 28 hex (RST 5). The routine at the label 'keybd:' then gets the character and places it in the keyboard buffer. The keyboard pointer and buffer count are both incremented, the interrupt flipflop is reset with an 'ei' command, and the computer returns to its other work.

When the computer needs another byte, it is obtained from the buffer by executing the code at label 'conin:'. The computer buffer pointer and buffer count are incremented at this time. Figure 1 shows the arrangement of the keyboard buffer and the pointers. While there is only one buffer, there are two pointers; one for the keyboard and one for the computer. There is also one byte which indicates the number of characters entered from the terminal and another which indicates how many characters the computer has read.

Whenever a carriage return is read from the buffer, the computer checks to see if both pointers are the same. If they are, both pointers are reset to the beginning of the buffer. This keeps the buffer from growing ever larger.

The buffered keyboard can be effectively used with ED, SID, DDT or Basic. With Basic, for example, the commands:

LIST <cr>
RUN <cr>

can be given, one after the other. Both will be executed in order.

Computer pointer EE00	Computer count EE02	Keyboard count EE03	Keyboard pointer EE04	BUFFER EE06

Figure 1. The input buffer and pointers

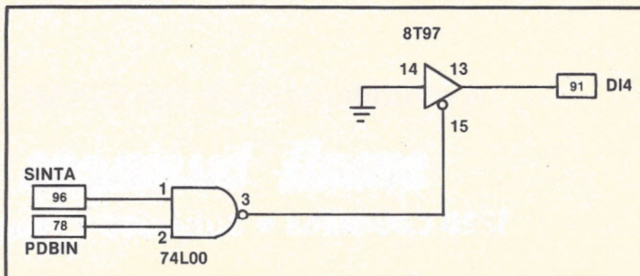


Figure 2. Circuit to convert an 8080 interrupt to an RST 5

Console output can be immediately suspended by typing a Control-S. Scrolling can be resumed by typing a Control-Q (unlike the usual CP/M setup where scrolling is resumed with any key). For CP/M programs such as SID, DDT or ED, the current task can be aborted by typing either a Control-C or a Control-E. (With regular CP/M, tasks are aborted by typing any key.)

Tasks in Microsoft or Tarbell Basic are aborted by typing a Control-C only. The Control-E option must be used for Xitan Basic. Neither a Control-C nor Control-E will immediately abort a task if the keyboard pointer is ahead of the computer pointer. In this situation, a Control-D must be typed first to empty the buffer. Then the task can be aborted with a Control-C or E.

Because CP/M disk operations are not usually interrupt driven, the computer interrupt-enable flipflop is disabled during disk access. This means that keyboard characters cannot be received during disk operations (whether a buffered keyboard is implemented or not). The original, single-density Lifeboat version does not re-enable interrupts after disk operations. But an NOP instruction was provided so that the user can patch in an 'ei' instruction. The initialization routine shown in the listing patches the Lifeboat version so that interrupts are enabled after each disk access.

Several CP/M programs such as DDT and SID use the RST 7 location at 38 hex. This makes it necessary to use a different restart location for the keyboard interrupt. But a single-level interrupt system on the 8080 requires this location.

One solution is to use a vectored interrupt board, so that restart levels 4, 5 or 6 can be chosen. The major disadvantage of this approach is cost: vectored-interrupt boards are several hundred dollars. Another approach is to utilize the circuit shown in figure 2. This method only requires two gates on two IC chips. The dual-input NAND 74L00 goes low when both the interrupt acknowledge line (SINTA) #96 and PDBIN #78 go high. Then the tristate buffer 8T97 is enabled so it can pull data-input line (DI 4) #91 low. This circuit, which may be added to the kluge area of an existing board, converts a simple interrupt from the keyboard to an RST 5 at address 28 hex.

Location of the buffer

The buffer must be capable of holding the longest line that might be typed. For a Basic interpreter that accepts several statements on a single line, the buffer should be able to hold at least 80 characters. This large a buffer cannot be placed in the CP/M user area between 40 hex and 100 hex. It would also be a disaster if the user inadvertently loaded a program over the keyboard buffer. Consequently, the buffer should not be placed in the CP/M transient program area above 100 hex. The best place for the buffer therefore is above CP/M itself.

For a North Star disk controller, the PROM operating system, located from E800 to EB00, places an upper limit on the CP/M system size. The address EE00 hex above the system PROMs then provides an ideal location for the input buffer. The buffer pointers can, of course, be located anywhere, but are placed at the beginning of the buffer for the program listing. □

A.R.M.

Program on Page 137

Accessing Assembly Language Routines from Basic

by Jon Lindsay

With the development of high level languages such as extended and disk Basics, the need for abandoning the expert managerial facilities of Basic is less than ever. The purpose of the interpretive language is to take care of the critical problems of where and how to use the instruction and data bytes of an operative program. Through the use of specific Basic commands and statements, data is easily manipulated from one computing framework to another.

The more I experiment with assembly language, the more I like its succinctness and precision. Yet, I write my serious programs in Basic for speed of programming and control. The issue is not whether a high level language is better than a lower level language. It is the recognition that the two have a place together, for those occasions when one becomes inadequate.

Special dedicated assembly language routines can be invaluable, such as math routines for rapid calculation and throughput. Quick string manipulations may be another incentive, as in dealing with a word processor when several strings must be concatenated. For instance, the Basic statement 'let A$(I) = L$ + C$ + R$' (where 'L$ = left(A$(I))', C$ = new character to be inserted, 'R$ = right$(A$(I))') may be barely acceptable. Yet, several text editors are written in assembly language and the result is very effective programming.

Initiating the process

Let's examine how to access such utilities while in the middle of an operating Basic program. Depending upon the Basic being used, the exact handling of those nasty bytes we have heretofore sought to avoid is probably a matter of the individual Basic. I will use MBasic by Microsoft with the CP/M operating system and the associated procedures. Version 1.4 supports the 'usr' command and the new version 5.0 will support both this and the 'call' command. Both will reach into memory and execute the routines then return to the main program.

In accessing an assembly language subroutine: (1) the subroutine itself can be called and executed; (2) arguments from the Basic program may be passed to the subroutine, operated on there, then passed back to the program. The second operation is a little more complicated and involves several registers. I will deal with the simplest form, the subroutine by itself.

I have written a simple routine that will print two messages (figure 1). Using an assembly language subroutine is not particularly easy to implement because getting it into memory requires some effort. We shall presume that the subroutine will work as constructed. The listing was assembled at 100H (hex) and tested. The listing for such a simple printing routine may seem a little strange because it makes use of the input/ output functions of CP/M. I could have used conventional pointers and standard input and output routines to accomplish the same thing. So the routine will have little meaning unless you are using CP/M.

```
                ;       THIS IS A TEST TO USE CP/M PRINT FACILITY
                ;
                ;
 8100                   ORG     8100H
 0005 =         BDOS    EQU     5
 8100 111181            LXI     D,PROMPT
 8103 0E09              MVI     C,9
 8105 CD0500            CALL    BDOS
 8108 112481            LXI     D,MESS1
 810B 0E09              MVI     C,9
 810D CD0500            CALL    BDOS
 8110 C9                RET
 8111 5468697320PROMPT  DB      'This is a PROMPT',0DH,0AH,'$'
 8124 5468697320MESS1   DB      'This is MESS1 $'
 8133                   END     8100H

A> DDT VERS 1.4

D8100
8100 11 11 81 0E 09 CD 05 00 11 24 81 0E 09 CD 05 00 ........$......
8110 C9 54 68 69 73 20 69 73 20 61 20 50 52 4F 4D 50 .This is a PROMP
8120 54 0D 0A 24 54 68 69 73 20 69 73 20 4D 45 53 53 T..$This is MESS
8130 31 20 0D 0A 24 00 00 00 00 00 00 00 00 00 00 00 1 ..$..........
```

Figure 1. Assembly language routine and its hexadecimal dump.

There appear to be two primary ways of entering your subroutine into memory. It could be done before your Basic is loaded. It is recommended that you reserve top user RAM for the subroutine that must reside somewhere in memory. In my subroutine, I arbitrarily picked 8100H (hex) or 33024 decimal as the starting address. Where one is going to use a relatively short program, reserving memory probably is not critical.

However, a Basic such as MBasic and a word processor are much different. With each additional line of text, the Basic will increment memory until it is depleted, overwriting the subroutine in the process. It is desirable to use the highest memory location option: A>MBASIC WORDPROC/M:40000. This loads Basic and the word processor and uses the first 40K of memory. In a 48K system, this leaves 8K for CP/M's FDOS and the assembly routine. More or less can be reserved as needed.

As memory is used, it is possible to test for remaining memory and take appropriate steps as it is depleted, without having to worry about destruction of the assembly routine. It has been suggested that a system monitor be used to place the routine in proper high memory. The most practical method for modest length routines is to 'poke' it into the proper memory slot, as I did here. In both examples, the data is placed in 'data' statements and 'read' (and poked) into memory. Don't be confused by the &H that precedes each number. This simply designates that the following number is hexadecimal. Short routines can be included with the main program. Longer routines should be located in separate programs, chained from the main program, and executed to memory. Then the main program can continue. The primary thing to note about the subroutine is the 'ret' mnemonic. This must be included in both 'usr' and 'call' methods in order to return to the main program.

```
10 PRINT CHR$(27);CHR$(43): REM  CLEAR SCREEN
20 PRINT "            THIS IS A TEST OF THE USR FUNCTION"
30 PRINT:PRINT:PRINT
40 PRINT "THE PROGRAM USR3 WILL BE SUMMONNED"
50 PRINT
60 I=&H8100:  REM  STARTING ADDRESS OF SUBROUTINE
70 READ A
80 IF A=&HFF THEN 120
90 POKE I,A:  REM  POKE BYTES INTO MEMORY
100 I=I+1
110 GOTO 70
115 REM ---------------------------------
120 DEF USR3=&H8100
130 X=USR3(1)
135 REM ---------------------------------
140 PRINT
150 PRINT "THE USR3 SUBROUTINE IS DONE"
160 DATA &H11,&H11,&H81,&H0E,&H09,&HCD,05,00,&H11,&H24,&H81,&H0E,&H09
170 DATA &HCD,05,00
180 DATA &HC9,&H54,&H68,&H69,&H73,&H20,&H69,&H73,&H20,&H61,&H20,&H50
190 DATA &H52,&H4F,&H4D,&H50
200 DATA &H54,&H0D,&H0A,&H24,&H54,&H68,&H69,&H73,&H20,&H69,&H73,&H20
210 DATA &H4D,&H45,&H53,&H53
220 DATA &H31,&H20,&H0D,&H0A,&H24,&HFF
```

Figure 2. Subroutine access via the 'usr' command.

In figure 2, the program first puts the 'data' statements into memory at the chosen location. In line 120, the starting address of the subroutine is specified. Ten different routines may be defined at any time. However, by simply redefining a subroutine's address, any number of 'def usr' statements may be used. In this example, the starting address of assembly language subroutine number 3 is at 8100H (hex). To summon the subroutine, refer to line 130. The format used is 'usr [<digit>] (argument)'. The digit is the subroutine number 3. The argument is the means by which data is passed to the subroutine. This example does not require an argument, so I gave it a simple number. However, several types of arguments may be used, i.e., 2-byte integer, string, single precision floating point number, and double precision floating point number.

Depending on which type is used, various memory locations in relation to the [H,L] register pair are precisely designated. Using the [H,L] register pair as pointers, data can be passed and retrieved at will. From line 130, returning data is usually of the same type as was passed. Here X contains the value to be returned. Again, we do not care what was returned; further processing can continue from this point. Line 150 prints another sentence and demonstrates that we have returned from the subroutine to continue the main program.

```
10 PRINT CHR$(27);CHR$(43): REM  CLEAR SCREEN
20 PRINT "            THIS IS A TEST OF THE CALL FUNCTION"
30 PRINT:PRINT:PRINT
40 PRINT "THE PROGRAM CALL WILL BE SUMMONNED"
50 PRINT
60 I=&H8100
70 READ A
80 IF A=&HFF THEN 120: REM  Is the message delimiter found?
90 POKE I,A
100 I=I+1
110 GOTO 70
115 REM ----------------------------
120 ADR=&H8100
130 CALL ADR
135 REM ----------------------------
140 PRINT
150 PRINT "THE SUBROUTINE IS DONE"
160 DATA &H11,&H11,&H81,&H0E,&H09,&HCD,05,00
170 DATA &H11,&H24,&H81,&H0E,&H09,&HCD,05,00
180 DATA &HC9,&H54,&H68,&H69,&H73,&H20,&H69,&H73
190 DATA &H20,&H61,&H20,&H50,&H52,&H4F,&H4D,&H50
200 DATA &H54,&H0D,&H0A,&H24,&H54,&H68,&H69,&H73
210 DATA &H20,&H69,&H73,&H20,&H4D,&H45,&H53,&H53
220 DATA &H31,&H20,&H0D,&H0A,&H24,&HFF
```

Figure 3. The 'call' command is implemented.

To use the 'call' command, refer to figure 3. This will require version 5.0 of Basic-80 (MBasic); version 1.4 does not support it. Substitute lines 120 and 130 of figure 3 for lines 120 and 130 of figure 2. The 'call' command appears to be somewhat easier to implement. Simply define an address and 'call' it. In this example, there is no argument so a simple 'call' (as in assembly language) will suffice. But a large amount of data can be passed via the 'call'. The format is 'call <variable name>[(argument list)]'. The variable name contains the address of subroutine (line 120). The arguments, depending on their amount, are passed first to the [H,L] registers, the second to the [D,E] registers, and three or more are pointed to by the [B,C] registers and form a contiguous block in memory.

```
                ;A SUBROUTINE TO PRINT A SIMPLE STATEMENT
                ;
 8100                   ORG     8100H   ;ASSEMBLE HERE
 8100 211481            LXI     H,MESSAGE       ;POINT REG. H TO MESSAGE
 8103 DB00      START   IN      00H     ;CHECK CRT READY STATUS
 8105 E680              ANI     80H     ;MASK READY BIT
 8107 C20381            JNZ     START   ;JUMP TO START IF NOT READY
 810A 7E                MOV     A,M     ;GET A BYTE OF MESSAGE
 810B D301              OUT     01H     ;OUTPUT CHARACTER TO SCREEN
 810D 23                INX     H       ;MOVE UP MEMORY LADDER
 810E FE00              CPI     00      ;IS IT THE END?
 8110 C20381            JNZ     START   ;IF NOT DO IT AGAIN
 8113 C9                RET             ;RETURN TO BASIC
 8114 415353454DMESSAGE DB      'ASSEMBLY LANGUAGE IS EASY!',00H
 812F                   END     8100H
*

DDT VERS 1.4

D8100
8100 21 14 81 DB 00 E6 80 C2 03 81 7E D3 01 23 FE 00 !.........~..#..
8110 C2 03 81 C9 41 53 53 45 4D 42 4C 59 20 4C 41 4E ....ASSEMBLY LAN
8120 47 55 41 47 45 20 49 53 20 45 41 53 59 21 00 00 GUAGE IS EASY!..
8130 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
8140
```

Figure 4. An assembly language routine and hex dump using more conventional I/O.

If you do not have any subroutine in mind but want to implement the 'usr' or 'call' commands now, figure 4 uses more common coding and can be used for experimentation.

Using an assembly language routine takes a fair amount of work and bother; it better be worth it. In our example, it certainly is not, but the example demonstrates the technique and should pique interest in testing this feature. □

NEW PRODUCTS

Property management software compatible with the TRS-80 model II is designed to meet the Institute of Real Estate Management recommended computer system capabilities. The software, suitable for all types of income properties, is a full general ledger program that keeps track of income and expenses providing formatted financial statements, management reports and exception reports. The software will run under CP/M and is also compatible with most Z-80 and 8080 systems. Price is $650.00. A-T Enterprises, 221 North Lois, La Habra, CA 90631, (213) 947-2762.
CIRCLE INQUIRY NO. 121

Inventory control system for the Apple II handles up to 8,100 items and has a transaction register and audit trails. The program generates inventory status reports, reorder reports and keeps track of purchase orders automatically. It also handles multiple departments or divisions, has fast data retrieval, and is easy to operate. Minimum hardware requirements are Apple II Plus with 48K, one disk drive and 80 column printer. Price: $99.00. Micro Business World, 15818 Hawthorne Blvd., Lawndale, CA 90260, (213) 370-4842.
CIRCLE INQUIRY NO. 122

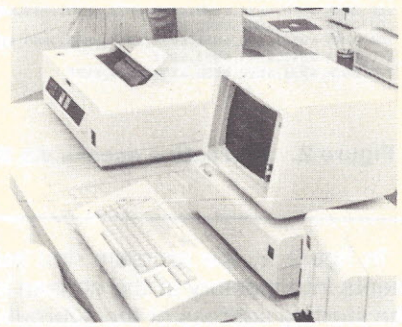
Report textwriter, Dart, software for the Apple II and Apple II Plus simplifies the creation of letters, reports, and other output text, utilizing disk files created by Edit. The system allows free-form file input, without regard to either line or page length, and assembles text into physical lines and pages using predefined operator or file parameters. Variable data may be key entered at report generation time utilizing standard text or form letter. File chaining allows an unlimited amount of input text. Other options include pagination, line justification, titles-subtitles, page numbering, line centering, margin setting, and double spacing. Price is $19.95. Services Unique Inc., 2441 Rolling View Dr., Dayton, Ohio 45431.
CIRCLE INQUIRY NO. 123

Floppy disk drives, series CBM 2030, disk drives designed for beginning computer users, have the lowest cost, smallest size and least weight of any drives currently available. With 130K bytes of storage, the unit can be field upgraded to a dual drive version which offers 260K bytes of storage. The drives use the same 5¼-inch media as Commodore's existing 340K CBM 2040 dual drive floppy disk for read-compatibility and operate on any Commodore PET or CBM. The cabinet, which can house two disk drives, measures 13½ × 8 × 6½ inches—smaller than most single-drive cabinets. A single drive model is $595. Commodore Business Machines, Inc., 3330 Scott Blvd., Santa Clara, CA 95051, (408) 727-1130.
CIRCLE INQUIRY NO. 124

Mail list merging option for Wordstar, Mailmerge, automatically incorporates data from other files containing names, addresses, special salutations and any other desired information, and automatically reformats the paragraph in which variables appear, to produce a perfectly spaced and marginated letter. Operating with Release 2.0 of Wordstar, it easily enhances options such as multiple copies and "chained" files for large document production. Price $150. Micropro International, 1299 Fourth St., San Rafael, CA 94901.
CIRCLE INQUIRY NO. 125

Integrated accounting system for CP/M based computers and Radio Shack products include general ledger, accounts receivable, accounts payable, and payroll which make up a complete business system. Extensive documentation is also provided. Programs are written in microsoft CP/M Basic and are available in source form for easy customizing. Price is $100 each or $350 for four. Omikron Systems, 1127 Hearst Ave., Berkeley, CA 94702, (415) 845-8013.
CIRCLE INQUIRY NO. 126

Electronic "dictionary," Displaywriter, is a low-cost text processing system that can check spelling of 50,000 words and communicate with other office equipment over ordinary telephone lines. The software-based system can divide the text into lines and pages and can create justified text.

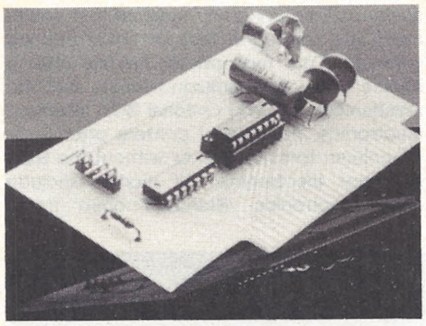


It has an electronic memory which provides up to 229,376 characters of information. Other options include three printers with speeds up to 15.5, 40 and 60 characters per second and a paper handler which moves each sheet into position and includes two paper trays. The printwheel printers use easy-to-change printwheel cartridges and ribbon cartridges as well as a ribbon which can print a million characters. Price is $7,895. IBM, Parson's Pond Dr., Franklin Lakes, NJ 07417, (201) 848-3454.
CIRCLE INQUIRY NO. 127

Report writer, designed to eliminate the process of compiling information into a ledger sheet, can be thought of as a matrix of rows and columns. Each element in the matrix may contain a literal string (title), a number, or a function (calculated number). Any element may be functionally related to other elements, e.g., element B5 may equal B3 divided by B2. Functions on a row or column may be replicated, such as a sales figure that begins in January and increases 10% each month through December. The format includes defining the number of rows

and columns; indicating whether or not a column entry will default to a literal string, decimal or integer number; defining the length of each column; and whether or not dollar signs, commas, or the % sign can be included in the hard copy. Under the data entry mode, data may be entered into a cleared format or a format with existing data. Data may be saved and recalled from disk. Price is $150. Carolina Business Computers, Inc. Oakwood Center, 350 Third Ave. NW, Hickory, NC 28601, (704) 328-3939.
CIRCLE INQUIRY NO. 128

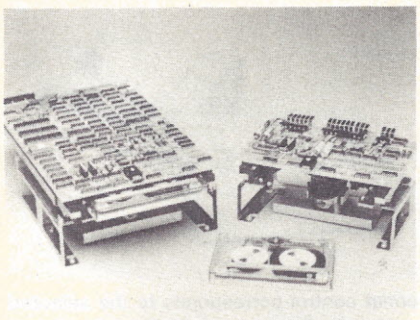
Low cost music synthesizer, Juke box, is designed to plug into any 48K Apple computer using Applesoft Basic. It is capable of producing three simultaneous voices and one channel of white noise. Pitch, rhythm, tempo, attenuation and envelope can be selected and controlled for each voice independently from other channels. It has a 5-octave range starting at 55Hz to 1760Hz.



Each card has an on-board amplifier capable of directly driving an 8-ohm speaker. A high level graphics music editor is included. The software enables music to be composed, edited, played and stored on a diskette. Cost is $129.95. American Micro Products, Inc., 705 N. Bowser, MS 107, Richardson, TX 75080, (214) 238-1815.
CIRCLE INQUIRY NO. 129

Low power cartridge system, Kennedy 6450, has both transport and formatter in one package that dissipates less than 40 watts. Designed to back-up 8-inch Winchester drives, the system with embedded formatter measures 8½ × 4½ × 16½ inches, allowing disk and transport to be placed side-by-side in a 19-inch rack. Using 6,400-bits-per-inch recording density and serpentine serial recording, the system has an unformatted capacity of 17.3M bytes with a 450-foot DC-300 cartridge and 11.5M bytes with a 300-foot cartridge. Transfer rate is 192K bits-per-second. Read/write speed is 30 inches-per-second and fast-motion speed is 90 ips. The microprocessor-based formatter has prewrite compensation to minimize pulse crowding and phase distortion for high write-data integrity. Group-coded recovery techniques supply 100 percent wider data recovery margins than conventional modified frequency modulation techniques. Price is $1,700. Kennedy Co., 1600 S. Shamrock Ave., Monrovia, CA 91016, (213) 357-8831.
CIRCLE INQUIRY NO. 130

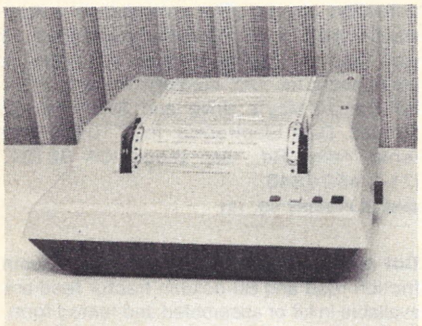
Streaming digital cartridge tape drives maintain 10 and 20 megabyte capacities and offer a choice of speeds—30 or 90 inches per second. This results in choice of transfer rates—1.67 megabytes-per-minute or 5 megabytes-per-minute. These drives are directed towards small, fixed



Winchester-technology disk back-up application. The microprocessor, streamer, is controlled with either a serial or parallel interface. GCR recording at 7,500 bits-per-inch is used with a variety of internal error detection checks. The error rate is a maximum $1:10^{10}$. A 4K I/O buffer is included, as well as an interface that simulates a semiconductor 'fifo' memory. Price is $746 for the 10-Mbyte drive and $788 for 20 Mbytes. Data Electronics, Inc., 10150 Sorrento Valley Rd., San Diego, CA 92121, (714) 452-7840.
CIRCLE INQUIRY NO. 131

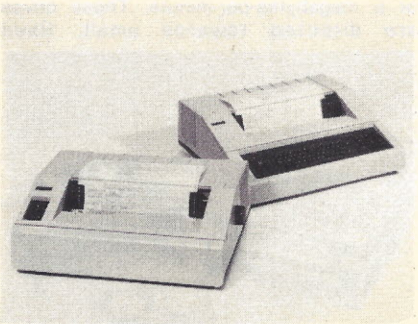
Dot matrix impact printers with 40-column friction feed and 80-column tractor feed are available in kit or assembled and tested form.

Features include: microprocessor controlled and programmable with 32 system-level software commands, graphics dot plotting mode, 96 Ascii characters with upper and lower case. Also, print with nine software



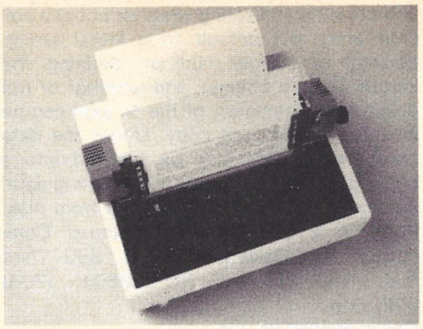
selectable sizes from 5×7 to 10×14 character fonts, reverse font printing capability, standard parallel and serial interface, 110 to 9,600 baud and adjustable tractor width for paper size slection. Prices for kits are $295 for the 40 column and $455 for the 80 column. Assembled and tested printers run $325 for 40 column and $485 for the 80 column. Coosol, Inc., P.O. Box 743, Anaheim, CA 92805, (714) 545-2216.
CIRCLE INQUIRY NO. 132

Dot-matrix serial printers offer refinements in paper- and forms-handling, five print pitches for greater flexibility, optional higher-density print matrices and RS-232-C interface. The HP 2631B printer and HP 2635B printing terminal allows paper to automatically advance to the next top-of-form whenever a paper-motion command causes text to fall outside the programmed text. The RAM-based 16-channel vertical



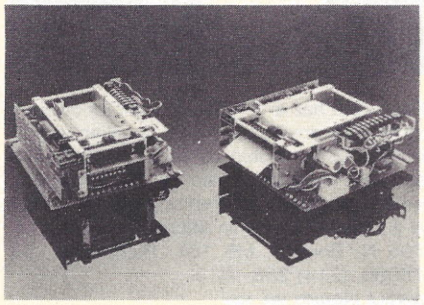
format control corresponds to the selected text length. Program-selectable print pitches available are: 12.5 characters per inch, 8.33 cpi, expanded 8.33 cpi, 6.25 cpi, and 4.16 cpi. The printers continue to offer 10 cpi for normal printing, and 16.7 cpi, making it possible to print 132 columns on paper 8½ inches wide, or 227 columns on 14⅞-inch (387 mm) paper. Price of HP 2631B is $3,600 and $3,950 for HP 2635B. Hewlett-Packard Co., 1507 Page Mill Rd., Palo Alto, CA 94304, (415) 857-1501.
CIRCLE INQUIRY NO. 133

80-column dot matrix printer, the TX-80, designed to interface with personal computers, is capable of printing a full 96 Ascii and 64 graphic characters at the rate of 125 cps. The printer uses a 5 x 7 matrix that allows the printing of descenders, and is controlled by an internal microprocessor in



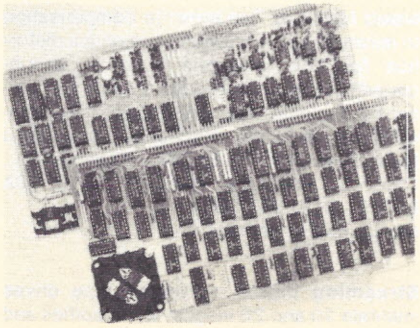
an all-metal cabinet. Optional interfaces include an RS-232 current loop, IEEE 488 and a custom Apple interface. It is available with either friction or tractor-type paper feed. Price with the standard friction-feed head is $710. Epson America, Inc., 23844 Hawthorne Blvd., Torrance, CA 90505.
CIRCLE INQUIRY NO. 134

Digital printers, designed for truck weighing scales, oil metering and medical test equipment, are offered in tape or ticket printing versions. Four tape printer models offer a choice of 6- or 12-column formats with the mechanical unit, and optional BCD interface electronics. Four ticket printers offer 5- or 11-column formats with or without the BCD interface electronics. All models include drive electronics. Standard open frame



units are supplied with taps for 12 v dc, 117 v ac or 220 v ac power inputs. Smith & Wesson, 2100 Roosevelt Ave., Springfield, MA 01101.
CIRCLE INQUIRY NO. 135

Graphics interface, SDI, designed for Cromemco computers, displays color or black-and-white images with up to 756-by 484-point resolution. Features are color map selection, dual page windowing function, automatic area fill mode and NTSC broadcast compatibility. Consisting of two circuit boards which plug directly into a S-100 bus, the interface uses direct memory access to display the contents of a display memory. Each pixel may be mapped from one nybble or one bit of the display memory. Bit-mapped



or nybble-mapped mode is software selectable. In nybble-mapped mode any 16 or 4,096 possible colors may be displayed in a single picture. In bit-mapped mode any two of these colors. For black-and-white nybble-mapped mode there can be 16 shades of gray. A bit-mapped black-and-white picture, on the other hand, yields strictly a black-and-white display. Price is $595. Cromemco, Inc., 280 Bernardo Ave., Mountain View, CA 94043, (415) 964-7400.

CIRCLE INQUIRY NO. 136

Z-80 based terminal, CPG-4010, operates in four modes to permit use for alphanumerics, graphics, or both. Compatible with Textronix Plot 10 software, it provides automatic scaling to allow use of compatible status read back that returns alpha cursor position and graphic status information upon receipt of an ESC ENQ. The terminal operates in ADM-3A Alpha, 4010 Alpha, vector and point modes. Allowing full screen usage, the Plot 10 graphic grid is automatically scaled to the CPG-4010 grid. Options include phosphorous green screen, hairline cursor control module, and plotter output. Price is $1,995. Continental Resources, Inc., 175 Middlesex Turnpike, Bedford, MA 01730, (617) 275-0850.

CIRCLE INQUIRY NO. 137

Data communications unit, Chatterbox, combines the COMM-80 I/O interface for the TRS-80 and an acoustic modem. The box, which is all that is required to turn a 4K TRS-80 into a full timesharing terminal, includes a built-in programmable 50-19,200 baud serial port, a Centronics compatible parallel printer port, a 300 baud acoustic

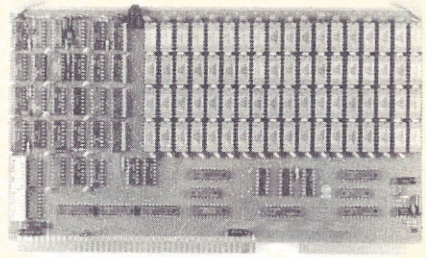
originate modem and a spare TRS-bus expansion connector. Also, power supply, connection cable, user's manual and smart terminal software are included. When the modem is in use, the complete data conversation is automatically routed to the serial output port where it can be logged on a printer. This is the only peripheral needed to allow a TRS-80 to communicate with timesharing systems such as Micronet or the Source. Cost is $259.95. Micromint Inc., 917 Midway, Woodmere, NY 11598, (516) 374-6793.

CIRCLE INQUIRY NO. 138

Double density controller board, for an SS-50 bus allows 336K bytes of storage on a single minifloppy and is capable of handling up to four 5¼-inch and four 8-inch drives simultaneously. The DCB-4 Disk Master, including DOS-68D or 69D, is fully compatible with hard disk systems and allows users to select either single or double-sided operation. The board occupies 16 bytes of memory space and can read and write a single sector by itself. An on-board buffer memory allows full interrupt capability in interrupt-driven systems, and one data transfer has been initiated, no further processor time is required. Extended decoding circuitry is incorporated for addressing the SS-50C bus, and includes disabling by jumper option. A phased-locked loop assures complete data integrity. Priced at $449. Smoke Signal Broadcasting, 31336 Via Colinas, Westlake Village, CA 91361, (213) 889-9340.

CIRCLE INQUIRY NO. 139

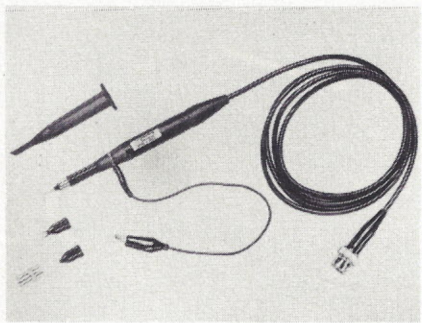
Memory module, CI-8086, available in 32K to 512K × 9 bytes on a single board, is compatible with both 8-bit and 16-bit multibus based systems. The memory is designed to plug directly into the backplane of any multibus compatible system. The module generates and checks even parity with selectable interrupt on parity error, and



allows maximum processor throughput with the use of on-board refresh control logic. Data access time is 250 nsec and cycle time is 375 nsec. The memory is addressable in 16K increments up to 16 megabytes. Prices: $1,500 for 128K × 9 and $8,700 for 512K × 9. Chrislin Industries, 31352 Via Colinas -#102, Westlake Village, CA 91361, (213) 991-2254.

CIRCLE INQUIRY NO. 140

Wideband instrument probe, model PR-40, is designed for use with oscilloscopes and frequency counters in applications through 100MHz. The slim-body probe with a 3-position switch selects a 10:1, a direct mode, or a reference position that grounds the tip through a 9M resistor. Accessories include a spring-loaded retractable tip-cover, a snap-on ground clip, BNC tip adapter, an IC tip



and an insulating tip designed for probing solid-state circuitry with no danger of shorting nearby components. For interface with test points or output jacks, the BNC adapter converts the probe tip into a push-on BNC connector. The IC tip guides the probe contact onto any pin of a standard DIP. Price is $34.00. Dynascan Corp., 6460 W. Cortland St., Chicago, IL 60635, (312) 889-9087.

CIRCLE INQUIRY NO. 141

Medical patient accounting system, Medpac II, a 64K microcomputer that handles all patient and third party billing, including printing insurance forms, provides control of all invoice data and is formatted for easy learning and operation. It records charges, adjustments and payments, and generates custom reports on services performed by each doctor in the practice. It stores data for up to 30 doctors and can print up to 40 insurance forms. It is sold as a turnkey system which includes delivery, set-up, and training for $14,249. V R Data Corp. 777 Henderson Blvd., Folcroft, PA 19032, (800) 345-8102.

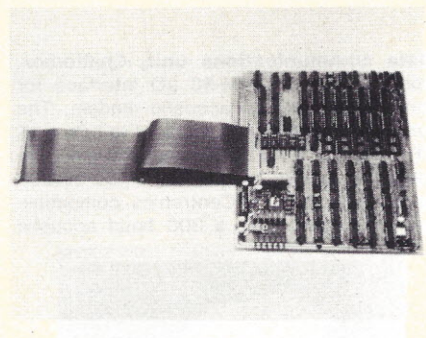
CIRCLE INQUIRY NO. 142

High performance graphics subsystems for all PDP-11 computers and the VAX-11/780 superminicomputer with full color raster-scan employs state-of-the-art microprocesser architecture to produce dynamic color displays. The subsystems will be aimed at general engineering, structural design and analysis, simulation, process control and monitoring, business graphics, and computer-aided design. Each model is available with either a 19-inch full color display terminal, the

VRV02, or a monochrome VT100 terminal. A joystick provides cursor control. Digital Equipment Corp., Maynard, MA 01754.

CIRCLE INQUIRY NO. 143

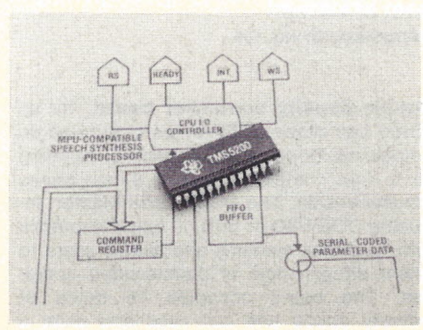
Graphic display board with high resolution Commodore PET computers provides video mixing and ROM sockets. The system is automatically restored to the user selected configuration after power-up or reset. The MTU K-1008-6 provides user control over a matrix of 64,000 dots (320 wide × 200 high). Serving as an 8K expansion memory when not used



for graphics, the board also creates a KIM/MTU expansion bus supported by various MTU products. On-board expansion allows use with optional light pen; K-1008-3C graphic software is also offered. Price is $320. Micro Technology Unlimited, 2806 Hillsborough St., Raleigh, NC 27605, (919) 833-1458.

CIRCLE INQUIRY NO. 144

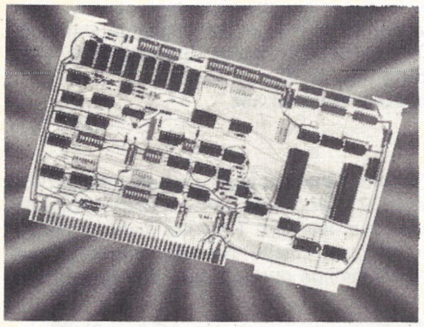
Synthesis processor, the TMS 5200, 'fifo' buffer is designed for easy interface to a standard microprocessor having an 8-bit data bus. It is currently used in the TI-99/4 home computer speech synthesizer peripheral and is suited for industrial and commercial applications. The speech chip has the capability of accepting data from either the TMS 6100, a compatible 128K-bit ROM



or the on-board 128-bit first-in, first-out buffer, which allows the storage of speech data in programmable read-only memories (PROMs), eraseable PROMs, random-access memory (RAM) or other random access storage media such as disk or bubble memories. Price is $100/100 pieces. Texas Instruments Inc., P.O. Box 1443, M/S 6404, (Attn: TMS5200), Houston, TX 77001.

CIRCLE INQUIRY NO. 145

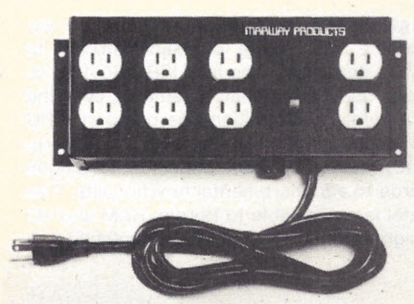
Supervisory board, SB-1004, provides features to reset a Multibus system after a malfunction, which is detected by a watchdog time-out technique. Should the system fail to resume normal operation after the reset, an on-board relay activates contacts which are available for utilization in system failure alarms. The board provides a reset



upon power-up, power brown-out and manual reset. To check and set parameters of the system, 24 user-definable status indicators and 24 user-definable input switches are provided. Also featured is a real-time clock interrupt with 6 user-selectable interrupt intervals from 100 microseconds to 10 seconds. Price is $395. Morrow Computer & Electronic Design, Inc., 315 Wilhagan Rd., Nashville, TN 37217, (615) 329-1979.

CIRCLE INQUIRY NO. 146

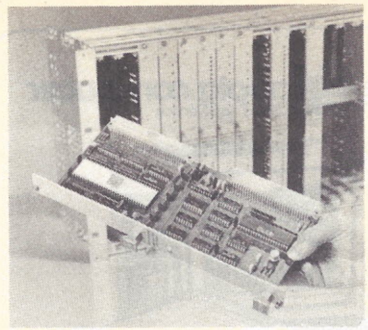
AC power controller, MPD 117, has eight outlets, EMI filter, built-in circuit breaker and other UL/CSA-approved components. It controls, filters and distributes AC power from a single outlet to up to eight devices for optimum performance of microcomputers, stereo and audio visual components and other office and home electronics devices.



All components are protected by metal fine-finish black chassis, with mounting flanges. Price is $89. Marway Products, Inc. 2421 S. Birch St., Santa Ana, CA 92707, (714) 549-0623.

CIRCLE INQUIRY NO. 147

CPU board, based on the 16-bit, TMS9900 microprocessor, provides the CPU functions for the 990E Industrial Microcomputer System for OEM and end-use applications. The CUP-200 provides a 56 line proprietary bus that allows memory addressing to 64K bytes and I/O capabilities. A total of 4096 inputs and outputs may be addressed individually. A serial port, provided on the module to be configured as either an RS232 or current loop, features a programmable baud rate of 75 to 38.4K baud, 5-8 bit character length and 1, 11/2 or 2 stop bits. Other features include: 16 vectored interrupts, 8-line-bit programmable port and an interval timer. The external control functions of the TMS 9900 are provided to the user. The compact 4-inch × 9-inch multi-layer board requires +5V @500mA, +15V @90mA and −15V @30mA. Power is applied to the module through the motherboard connector. The 3 MHz clock rate provides a



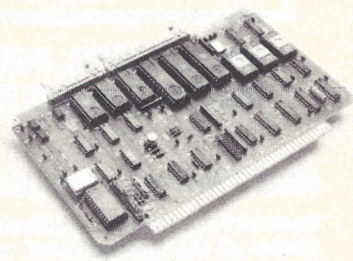
2 microsecond 'branch', and a 31 microsecond 'divide'. Price is $560. Erni & Co., 3316 Commercial Ave., Northbrook, IL 60062, (312) 480-9240.

CIRCLE INQUIRY NO. 148

AC power conditioners, W-model voltectors, remove electrical disturbances from the power line to protect micro/minicomputers, word processors and other sensitive electronic equipment from noisy and unstable AC power. The power conditioners reduce excessively high voltage surges to safe levels. They also remove low voltage transients and noise spectra that often cause computer errors, data and program loss. They are effective in both common mode (line-to-ground) and transverse, or differential mode (line-to-line) forms of interference, smoothing out the 60Hz waveform to a relatively clean sine wave. Internally fused, the conditioners, equipped with an on/off switch and a green pilot light, are contained in a lightweight housing. Pilgrim Electric Co., 29 Cain Dr., Plainview, NY 11803, (516) 420-8989.

CIRCLE INQUIRY NO. 149

Single board microcomputer, the 9600A MPU module, is a complete MC6802-based microcomputer on a single card. The unit is supported by a family of modules for memory and I/O expansion and is bus and outline compatible with the Motorola Exorciser and Micromodules. Some features include: 1152 bytes of static RAM; 2048 bytes of Eprom, expandable to 6144 bytes; full duplex RS232C serial I/O channel, expandable to two channels; address and data bus fully buffered for external expansion; resident system monitor provides debug assistance; power failure protect/restart circuit; and break-detect circuit to generate reset or interrupt. Price is $595. Creative Micro Systems, 11642-8 Knott Ave., Garden Grove, CA 92641, (714) 898-9669.



CIRCLE INQUIRY NO. 150

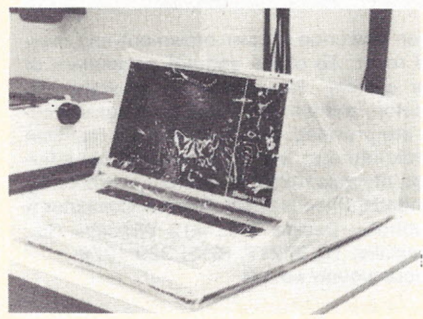
Catalog programs in ROM for the Apple II are designed to be installed on Mountain Computer's card and expands the capability of the Apple to include special copy programs and catalog information. A few of the commands include: identify disk volume number and amount of unused sectors on diskette; determine active drive or change active drive; 'catalog' gives name of file, length expressed as number of sectors, file type, and for binary files - stored load address and length expressed as bytes - either decimal or hexadecimal. Copyrom, a 2K ROM provides access for frequently used programs. Mountain Computer Inc., 300 Harvey W. Blvd., Santa Cruz, CA 95060, (408) 429-8600.

CIRCLE INQUIRY NO. 151

S-100 compatible DAC card offers 64 output channels from a single multiplexed 8-bit DAC. The system's processor accesses each channel individually without preliminary set-up sequences. Features include: C-10 volt range; 5-bit nonlinearity; 9-bit bus comparator for address decoding; 8 microsecond refresh cycles; low-cost high-density. The Model SB64-R is $539. Digital Multi-Media Control, 1210 E. Pine St., Stayton, OR 97405, (503) 769-2735.

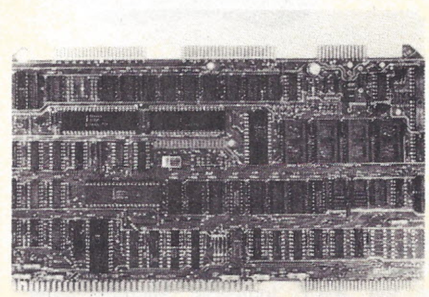
CIRCLE INQUIRY NO. 152

Terminal dust covers are designed to protect CRT terminals, printers and keyboards from dust and dirt. They are made of heavy gauge clear plastic that will protect against water damage during a fire. The covers are custom made to fit a specific model of computer CRT, keyboard or printer. Prices: CRT (including keyboard): $9.95; keyboard only: $8.95; printer: $9.95. Computer Accessories Co., 20 Boat Lane, Port Washington, NY 11050, (516) 767-0366.



CIRCLE INQUIRY NO. 153

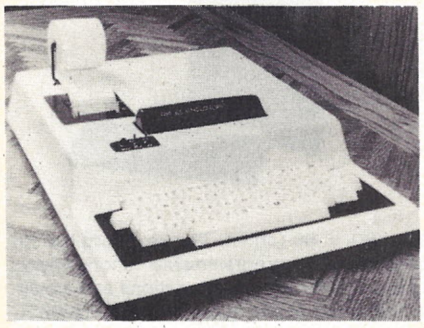
Single-board computer includes an expandable I/O feature of major interest to system designers. The Intel iSBC 80/10B includes specification improvements over the /10A version. In addition, the on-board CPU bus has been extended to receive an entirely new family of iSBX Multimodule plug-in boards to add incremental functionality. The model is expandable to Eprom, RAM and I/O design needs for a built-in adaptability. A



choice of four types of Eprom/ROM (2708, 2758, 2608, 2716 and 2732) allows expansion from 4K to 16K bytes; 1K of RAM is included and may be extended up to 1K using Intel standard 2114A-5 1K × 4 static RAMs (sockets for these additional RAMs are provided on the ISBC 80/10B board). Price: $560. Intel Corp., 5200 N.E. Elam Young Parkway, Hillsboro, OR 97123, (503) 640-7147.

CIRCLE INQUIRY NO. 154

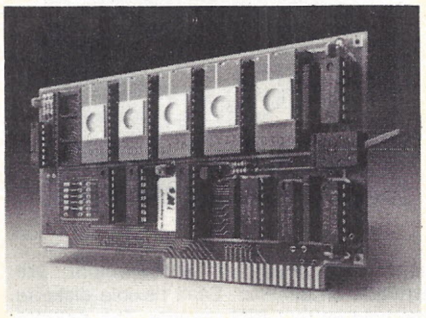
Five-component system lets designers quickly prepare custom circuit boards for S-100-bus systems without costly and time-consuming photo-negative processing. Heart of the system is a positive-resist-coated, double-sided copper-clad circuit board, model 8800R2. The 5.3 by 10 by 0.062 inch FR4, G10 epoxy-glass board is form and bus compatible with the S-100 convention. At the lower edge are 100 gold-flashed, nickel-plated card-edge contacts (50 each side) on 0.125 inch centers. The



contacts continue into the two-ounce copper fields so that no jumpers are required after etching. Both sides of the board are precoated with positive photo-resist. Provided with the board are layout paper, clear mylar film for artwork, a heavy plastic bag for etching and complete instructions. The 8800R2 boards are $19.95 each; the R407 transfer artwork kit, $2.65; the 259 etchant, $1.69; the 2594A etchant, $1.30; and the 0088-21-45 developer, $2.46. Vector Electronic Co., 12460 Gladstone Ave., Sylmar, CA 91342, (213) 365-9661, TWX 910-496-1539.

CIRCLE INQUIRY NO. 156

6502 development system contains within one program a text editor, assembler, disassembler and symbolic debugger. The system is available on Apple compatible diskette and resides in slightly over 8K RAM.



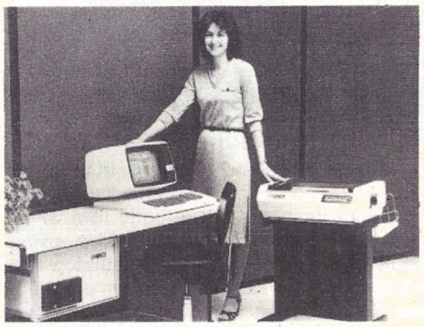
It is also available on Eprom with socket adaptors to replace Integer Basic and convert the Apple to a dedicated 6502 development tool. Microproducts, 30420 Via Rivera, R.P.V., CA 90274, (213) 541-5131.

CIRCLE INQUIRY NO. 157

Home builders business system, a complete, turnkey, microcomputer system consists of a North Star Horizon II computer, video terminal, and printer. The system does complete job costing—allowing each house under construction to be broken down into up to 99 user definable cost center. The system also provides a fully integrated general ledger to keep track of accounts payable to vendors and subcontractors, and provides financial statements. Dalcon Intl, 511 Woodbine Ave., Nashville, TN 37211, (615) 242-5801.

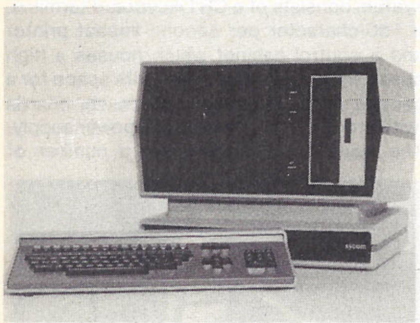
CIRCLE INQUIRY NO. 158

Advanced business computer features a unique, decentralized system architecture, entirely modular in design. It can be expanded from a basic, low-cost system to one utilizing more than 32 peripheral devices and approaching large, expensive computers in capabilities and storage capacity. The basic system consists of a CRT/keyboard terminal, a 150-character per second impact printer and a control cabinet which houses a high speed floppy disk dual drive (with space for a second drive), two plug-in processor boards (with provision for a third) and a power supply. The system is available with a number of



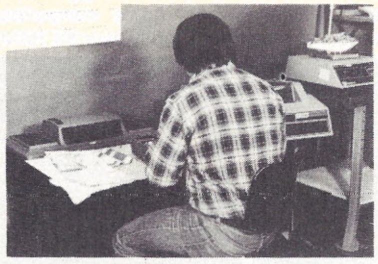
complete, field-proven software packages. The system runs all software developed for earlier Infotecs computer systems such as general accounting, fuel oil, wholesale/retailing, insurance agency accounting, job cost control, order entry and inventory control. Infotecs, One Perimeter Rd., Manchester, NH 03103, (603) 624-2700.
CIRCLE INQUIRY NO. 159

Microhost/development system for multiple uses in the difficult industrial environment, the 3805A, is a stand-alone, floppy disk based unit for software development and distributed network applications. With processing power supplied by a Z80A microprocessor, the system includes 96K of RAM (no wait states and bank switched memory



management), parallel and serial I/O, and integral 12 inch CRT, a free standing 93-key intelligent keyboard, an integral IBM 3740 format compatible floppy disk drive giving 250K of mass storage, and a tabletop line printer. Options include an Eprom programmer and a single, dual or triple disk drive floppy subassembly providing up to 1Mb of mass storage. Prices are below $10,000. Xycom, Inc., Box 984, Ann Arbor, MI 48106.
CIRCLE INQUIRY NO. 160

Receiving/inspection system Tab 7080 consists of a key-to-diskette data collection and retrieval terminal with built-in software for receiving and incoming inspection applications. Optional communications facilities are also available. Reports such as open purchase orders, closed purchase orders, parts awaiting inspection, parts completed inspec-



tion, and inspector achievement are available on demand. Each flexible diskette can hold up to 1,898 records. The diskette conserves file space and allows sharing data among departments. Diskettes can be mailed or hand delivered, or the information can be transmitted over the telephone. Another advantage is the elimination of guesswork about a vendor's reliability. With a quality history of a vendor on record, a comparison can be drawn by looking at the acceptance rate versus the rejection rate for a specific part. Price: $12,925. Tab Products, 1451 California Ave., Palo Alto, CA 94304, (415) 858-2500.
CIRCLE INQUIRY NO. 161

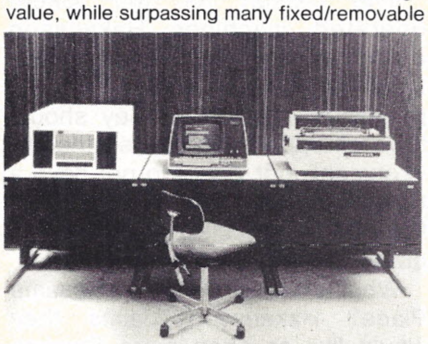
Information processor model 675 provides more than 2.6 million characters of diskette storage in a compact desk-top configuration. With high speed 655,360 character, 5.25-inch disks, the two versions are called Fast-Track and Fast-Track 2 + 2 (four disks). The 675 has a large 12-inch video display with a capacity of 20 lines, each with 80 oversized characters. It can be fitted on a plug-in basis with a model 60 full-page video display, which has 4,800 character capacity

in an 80 × 60 format. The keyboard has 20 separate user programmable function keys to provide 60 separate functions and a built-in asynchronous/synchronous EIA RS-232C communications interface. An extensive software library is available, including all existing Compucorp word processing and

data processing software. Prices: Fast-Track—$9,150; Fast-Track 2+2—$11,500. Compucorp, 1901 S. Bundy Dr., Los Angeles, CA 90025, (213) 820-2503, TWX: 910-3427578.
CIRCLE INQUIRY NO. 162

Text editor 2001E with 128k memory features a powerful Zilog Z80A microprocessor providing the system with 128k bytes of memory. By doubling the memory to 128kb, the system permits integration of various software programs on one module diskette, as well as development of special application packages for specific working environments. It enables the operator to design programs, using a Micom language or Basic. The 128kb memory is also being offered as an option to Micom users who want to update their present system to 128kb. Micom will field upgrade any 2000 to 2001E capabilities. Price: $17,900. Micom Data Systems, Inc., Suite 100, 4040 McEwan, Dallas, TX 75234, (214) 386-5580.
CIRCLE INQUIRY NO. 163

Office information system model OIS 105 is designed for small office clusters where functionality required by a first-time or sophisticated user is distributed to the point of need. The model offers increased software performance and new hard disk technology at a low price. The system incorporates new fixed disk drives for cost/storage value, while surpassing many fixed/removable



disks in performance. The hard disks are physically located within the master, so the compact system blends easily into any office environment. The completed Wang OIS family now includes the models 105, 115-1, 115-2, 125A, 130A, 140 and 145. Price begins at $9,300, U.S. Wang Laboratories, One Industrial Ave., Lowell, MA 01851, (617) 459-5000.
CIRCLE INQUIRY NO. 164

File management system FMS-80 allows the user to program applications in less time than normal with higher level languages like Basic. It is completely menu driven and steps you through setting up file definitions, select requirements, and screen and report definitions. Written in assembly language, the program is extremely fast and runs on many CP/M and MP/M operating systems or Cromemco's Cdos. FMS-80's features include the capability of screen formatting, report generation, multiple file handling, and arithmetic manipulation of multiple data records. There is no limit to the size of records or number of fields inside the files. Systems Plus, 1921 Rock St., Suite #2, Mt. View, CA 94043, (415) 969-7047.
CIRCLE INQUIRY NO. 165

Non-technical word processor, Wordstar, has been enhanced with several features, including installation program, mailing list merging, directory display and foreign language prompting. The screen-oriented, CP/M compatible, integrated system enables full-speed 1250 baud printer operation and interfaces with most terminals and printers without patching. Flexible formatting allows a variety of complicated mail list merging functions. Directory display permits scrolling text and CP/M file directories independently on a triple split screen. Foreign language translated versions of menus and messages are made possible by having all operator prompt messages and menus contained in a separate user-editible file on diskette. Other capabilities include using the CRT screen as a movable window into the text, split screen help menus, powerful editing commands, simultaneous editing and printing, justification, and automatic moving of text between disk and CPU for handling large documents. Video edit functions operate on any CRT or video board with cursor positioning functions and CP/M console device accessibility. Software: $495; mail merge option: additional $150. Micropro Intl., 1299 4th St., San Rafael, CA 94901, (415) 457-8990.
CIRCLE INQUIRY NO. 166

Keyed file management system to run on I.5 or II.0 UCSD Pascal systems allows writing programs which access data using user defined keys. The system PFAS uses a B+ tree file organization to ensure the time required to find information does not change with the number of records in the file. The standard version, using less than 6K bytes, can: make a new file, open an existing file, add records to a file randomly or sequentially by key, locate and retrieve records randomly or sequentially by key, start reading sequentially with any record and change records already in the file. The extended package also provides real-time record and key deletion and retrieval of records by partial key. Both versions are available in machine readable form on a 8-inch soft sectored, single or double density floppy disk. A user manual is included. Prices: Standard: $100, Extended: $150, Both: $175. C.J. Wigglesworth Software, P.O. Box 755, Cardiff, CA 92007.
CIRCLE INQUIRY NO. 167

Collection of interactive accounting software, Order Entry, will handle the documentation and control of both purchasing and sales. Further, the relevant information will travel through the other programs—accounts payable, accounts receivable, inventory control and general ledger—updating those modules to reflect purchase and sales activity. Capabilities include: generation and printing of purchase orders and sales orders in mailable format, computation of tax on taxable purchase orders and sales orders, registration of deliveries against outstanding purchase orders and of shipments against outstanding sales orders, complete purchase order and sales order history reports. Available in Micropolis 1053/II (48K), Apple II, PET (DOS 2.0) and Microsoft under CP/M. Compumax, P.O. Box 1139, Palo Alto, CA 94301, (415) 321-2881.
CIRCLE INQUIRY NO. 168

Sesame Place

Continued from Page 15

with fun and eliminate the confusion a child feels when he learns to operate a computer. Hakansson, who began her career with computers at the Lawrence Hall of Science at the University of California, Berkeley, has been working on the Sesame Place project for almost two years.

"I felt it was important that, before a whole new generation grew up in fear and intimidation, it needed hands-on experience with computers. The more I saw children use computers and interact with them, the more convinced I became that it was a magnificent way to teach."

Hakansson goes on to explain that even the most patient teachers place expectations upon their students. "A student feels a lot of stress and pressure when dealing with the adult community—he always feels the need to perform and meet up to the teacher's expectations. You take that away when you allow the child to work with a computer, which deals with him on an individual basis. The computer—being non-judgemental—lets a child take as long as he needs to answer a question."

While many think of a computer as a cold, complicated machine, Hakansson feels that it can generate a very personal experience, especially in programs that deal with each user on an individual basis.

Types of games

Some games in the gallery are designed to aid a visitor in developing hand-and-eye coordination, another in logical strategies, or skills in reading. Perhaps a player would like to try a music game or a creative writing exercise. For example, youngsters use the computer to write their own Sesame Place mystery. If they don't like that, they may want to try and guess the patterns forming Oscar, Big Bird, Cookie Monster and the rest of the muppets. A business-oriented game shows a young entrepreneur operating a lemonade stand what decision-making choice he must make to stay profitable.

Other samples include a game which has a colorful array of letters march on the screen to form a jumbled word. The player has to unscramble the letters before they march into their correct position. Another game, the Face Maker, uses the computer as a creative tool. The player chooses from various facial shapes and features to create unique characters.

The majority of the machines have a menu of two to three programs. But there are also dedicated machines, running one program since some games access a lot more files.

All the computers are token-operated with the games lasting approximately 4 minutes. "It's a friendly 4 minutes," Hakansson says. "Rather than cutting you off, the timing mechanism will check and, if the player should hit a natural ending a few seconds after 4 minutes that's fine; if he hits it before, that's fine too. It allows you to complete the exercise."

The reason for the token-operated machines Hakansson claims, is more for crowd control than anything else. "We feel that if people pay to get in they should have an opportunity to use the environment, but if we do not have a token operation, visitors will sit in front of the terminals for hours, not giving others an opportunity."

What are future plans for Sesame Place? Hakansson claims they would like to open more Sesame Places around the country. "The more we expand, the more children are exposed to 'technological' learning. Computers should be another resource in the school environment and Sesame Place is one step towards making the public more aware."

Hakansson hopes to mass market the software developed for Sesame Place next year. She wants to integrate it with Sesame Street's learning concepts. "Then a child at home could watch Sesame Street, turn off the TV, turn on the computer and play games, drawing on the ideas he just saw on TV.

"As the micro population explodes and becomes more affordable, parents are going to start asking whether their children are using computers in schools. I believe they will make choices dependent on that answer."☐

Now NRI takes you inside the world's most popular microcomputer to train you at home as the new breed of computer specialist!

NRI teams up with Radio Shack to teach you how to use, program and service microcomputers...make you the complete technician.

It's no longer enough to be just a programmer or a technician. With microcomputers moving into the fabric of our lives (over 200,000 of the TRS-80™ alone have been sold), interdisciplinary skills are demanded. And NRI can prepare you with the first course of its kind, covering the complete world of the microcomputer.

Learn At Home in Your Spare Time

With NRI training, the programmer gains practical knowledge of hardware, enabling him to design simpler, more effective programs. And, with advanced programming skills, the technician can test and debug systems quickly and easily.

Only NRI gives you both kinds of training with the convenience of home study. No classroom pressures, no night school, no gasoline wasted. You learn at your convenience, at your own pace. Yet you're always backed by the NRI staff and your instructor, answering questions, giving you guidance, and helping you over the tough spots.

Training includes TRS-80 computer, transistorized volt-ohm meter, digital frequency counter, and the NRI Discovery Lab with hundreds of tests and experiments.

(TRS-80 is a trademark of the Radio Shack division of Tandy Corp.)

Explore the TRS-80 Inside and Out

NRI training is hands-on training, with practical experiments and demonstrations as the very foundation of your knowledge. You don't just program your computer, you introduce and correct faults ...watch how circuits interact...interface with other systems...gain a real insight into its nature.

You also build test instruments and the NRI Discovery Lab, performing over 60 separate experiments in the process. You learn how your trouble-shooting tools work, and gain greater understanding of the information they give you. Both microcomputer and equipment come as part of your training for you to use and keep.

Send for Free Catalog... No Salesman Will Call

Get all the details on this exciting course in NRI's free, 100-page catalog. It shows all equipment, lesson outlines, and facts on other electronics courses such as Complete Communications with CB, TV and Audio, Digital Electronics, and more. Send today, no salesman will ever bother you. Keep up with the latest technology as you learn on the world's most popular computer. If coupon has been used, write to NRI Schools, 3939 Wisconsin Ave., Washington, D.C. 20016.

NRI Schools
McGraw-Hill Continuing Education Center
3939 Wisconsin Avenue
Washington, D.C. 20016
NO SALESMAN WILL CALL
Please check for one free catalog only.

All career courses approved under GI Bill.
☐ Check for details.

- ☐ Computer Electronics Including Microcomputers
- ☐ TV/Audio/Video Systems Servicing
- ☐ Complete Communications Electronics with CB • FCC Licenses • Aircraft, Mobile, Marine Electronics
- ☐ CB Specialists Course
- ☐ Digital Electronics • Electronic Technology • Basic Electronics
- ☐ Small Engine Repair
- ☐ Electrical Appliance Servicing
- ☐ Automotive Mechanics
- ☐ Auto Air Conditioning
- ☐ Air Conditioning, Refrigeration, & Heating including Solar Technology

Name (Please Print) Age

Street

City/State/Zip

Accredited by the Accrediting Commission of the National Home Study Council 171-100

Free Literature

● **Word processors.** A 4-page booklet details the RIII series of display text editors. Three models for various levels of editing applications are included: System 320, for queued printing, content search and automatic letter writing; System 330, with time-saving features and productivity report capability; and 340, providing selective retrieval and forms mode capabilities. Burroughs Word Processing Systems, 95 Horse Block Rd., Yaphank, NY 11980. **CIRCLE INQUIRY NO. 200**

● **Connectors.** An 18-page catalog offers information on UBS and UBC series crimp contacts, including material and finishing data, mechanical features, and electrical specifications. ITT Cannon Electric, 666 E. Dyer Rd., Santa Ana, CA 92702, (714) 557-4700. **CIRCLE INQUIRY NO. 201**

● **Educational books.** List of current books about computers for elementary school children is offered. A resource for parents and teachers who want books about computers before sitting children down to work with a keyboard. Bayshore Books, P.O. Box 848-B, Nokomis, FL 33555, (813) 485-2564. **CIRCLE INQUIRY NO. 202**

● **Intelligent data storage.** Brochure describes products and applications for microprocessor-controlled, RS-232 based cartridge, disk and solid state buffer storage systems. Columbia Data Products, 9050 Red Branch Rd., Columbia, MD 21045, (301) 992-3400. **CIRCLE INQUIRY NO. 203**

● **Cartridge tape sub-systems.** Product guide for popular mini-microcomputer architecture. Contains specifications on emulations, min./max. record lengths, storage capacities, transfer rates, power requirements, and prices. Alloy Engineering Co., Ron Richard, 85 Speen St., Framingham, MA 01701, (617) 620-1710. **CIRCLE INQUIRY NO. 204**

● **NC codes listed.** Pocket-sized numerical control chart lists codes in EIA and Ascii configurations. The EIA code consists of 63 characters including numerals, letters, and symbols; the Ascii code, 128 characters. Ron Wollman, Dept. P-60, ABR Corp., 262 Mott St., New York, NY 10012, (212) 431-7820. **CIRCLE INQUIRY NO. 205**

● **Computer tabletops.** Data bulletin describes the characteristics of Norcore plastic honeycomb for the needs of computer manufacturers. Data bulletin No. 164, Norfield Corp., 36 Kenosia Ave., Danbury, CT 06810. **CIRCLE INQUIRY NO. 206**

● **OEM and resellers guide.** A 1980 edition of the Microage Wholesale Price Guide provides manufacturers' suggested list prices and warranty policies for over 500 microcomputer products from more than 60 manufacturers. Microage Wholesale, 1425 W. 12th Pl., Tempe, AZ 85281, (800) 528-1415. **CIRCLE INQUIRY NO. 207**

• **Breadboard component kits.** A 12-page booklet details general purpose equipment for R&D layout work, prototype development, and production design of instruments and mechanical mechanisms. Assembly information is included. Winfred M. Berg, 499 Ocean Ave., East Rockaway, NY 11518. **CIRCLE INQUIRY NO. 209**

• **Digital switch units.** A 4-page brochure describes the functions of the DSU-2400 series, including typical applications and a configuration guide. Other info included: rack mounting and gang switching techniques, lists of digital switch units, rack adapters, cables, and accessories. Spectron, 344 New Albany Rd., P.O. Box 620, Moorestown, NJ 08507, (609) 234-5700. **CIRCLE INQUIRY NO. 210**

• **Data conversion components.** A 250-page catalog details line of products and tutorials on A/D, D/A converter specifications, including data sheets, notes for military and commercial applications, and tech briefs. Micro Networks Co., 324 Clark St., Worcester, MA 01606, (617) 852-5400, TWX (710) 340-0067. **CIRCLE INQUIRY NO. 211**

• **MOS/LSI devices.** A 256-page catalog includes data communication equipment, CRT devices, printers, baud rate generators, keyboard encoders and microprocessor peripherals. Standard Microsystems, Sales Dept. 35 Marcus Blvd., Hauppauge, NY 11787, (516) 273-3100. **CIRCLE INQUIRY NO. 212**

• **Technical journal.** A 20-page catalog features applications of 3-port isolation, digital-to-4-20mA converters for process control, a low-cost programmable measurement and control system, and CMOS D/A converters. New products include a 12-bit, 5 MHz A/D, a 10-bit 15ns D/A, an 8-bit low cost D/A, a 1.2V reference, a Bifet Op Amp. Analog Devices, Rt. 1 Industrial Park, P.O. Box 280, Norwood, MA 02062, (617) 329-4700. **CIRCLE INQUIRY NO. 213**

• **Tax forms guide.** A 28-page brochure provides complete information on IRS-approved forms for tax reporting via paper or magnetic media. Featured are illustrations of W-2, W-2P, 1099 and other stock tax forms positioned over true-to-size grids for use in programming computer printouts. Also shown are new open year W-2, W-2P, and interest forms for magnetic tape reporting, continuation sheets, and datamailer forms to eliminate inserting, sealing and metering. Uarco Inc., W. County Line Rd., Barrington, IL 60010. **CIRCLE INQUIRY NO. 214**

• **Short-form product guide.** A 30-page booklet describes Intersil line of analog, digital, horological and microprocessor integrated circuit products. Devices are fabricated with CMOS/LSI, MOS/LSI, bipolar LSI, and a proprietary process relating to vertical MOS power FETs. Also included are data acquisition components and systems; memories, microprocessors, and development systems; multiplexers and switches; and discrete MOS and bipolar transistors. Intersil, Inc. 10710 N. Tantau Ave., Cupertino, CA 95014, (408) 996-5000, TWX 910-338-0171. **CIRCLE INQUIRY NO. 215**

CALENDAR

Oct 4-5 Tidewater Hamfest/Computer Show/Flea Market, Virginia Beach, VA, technical forums and exhibitions of equipment for computer and radio applications. Tidewater Radio Conventions, P.O. Box 7101, Portsmouth, VA 23707.

Oct 7-10 International Congress for Data Processing and Software Exchange, Int'l. Congress Centre Berlin, trends in data processing, development and problems of modern hardware, innovative applications in systems, computers and industry and educational tasks in computer age. AMK Berlin, Messedamm 22, D-1000 Berlin 19, W. Germany, (030) 30 38-1.

Oct 8-10 Circulation Computer Systems Symposium, Marriott Hotel, Chicago, IL, selection of hardware and software, new product support, mail room and distribution controls, customer service, subscriber/non-subscriber files, geared for current or potential users of circulation computer systems. American Newspaper Publishers Assoc., Box 17407, Dulles Int'l. Airport, Washington, D.C. 20041.

Oct 8-22 Electronics Tour, Korea, Japan, Taiwan and Hong Kong Electronics Shows, develop foreign markets, observe foreign technology and innovations, seek foreign capital and investment, develop new products and improve personal contacts with foreign counterparts. Commerce Towers Int'l., 870 Market St., Suite 762, San Francisco, CA 94102, (415) 433-3072.

Oct 9-10 Museum Computer Network Annual Conference, New York State Museum, Albany, NY, summarizing the state of computer applications in US and Canadian museums. David Vance, Museum Computer Network, Library E-2340, State Univ. of NY, Stony Brook, NY 11794, (516) 246-6077.

Oct 14-16 Mini/Micro Conference and Exposition, Civic Auditorium, Brooks Hall, San Francisco, CA, technical program and product expo devoted to small computers. Mini/ Micro Computer Expo, 32302 Camino Capistrano, Suite 202, San Juan Capistrano, CA 92675.

Oct 14-17 Australian Technology Resources Exchange Fair, TRS -80, Victorian Government Pavilion, R.A.S. Showgrounds, Melbourne, Australia, exhibition/conference covering energy management and alternative, renewable energy sources, innovations in production processes, microprocessors, scientific and laboratory instrumentation. TMAC, 680 Beach St., Suite 428, San Francisco, CA 94109.

Oct 16-19 Mid-West Computer Show, McCormick Place, Chicago, IL, end-user exhibition featuring small and medium-sized business systems, scientific, engineering computers, and microcomputers. National Computer Shows, 824 Boylston St., Chestnut Hill, MA 02167, (617) 739-2000.

Oct 30-Nov 1 National Small Computer Show, New York Coliseum, New York, NY, hardware displays of minis, micros, peripherals, business and industrial software. National Small Computer Shows, 110 Charlotte Pl., Englewood Cliffs, NJ 07632.

Nov 13-16 International Computer Music Conference Queens College, Flushing, NY, concerts, workshops, panel discussions, demonstrations, exhibitions. Dr. Hubert S. Howe, Jr., Queens College, Flushing, NY 11367.

Nov 20-23 Northeast Computer Show, Hynes Auditorium/ Prudential Ctr., Boston, MA, business systems, scientific and engineering computers, and variety of micros. National Computer Shows, 824 Boylston St., Chestnut Hill, MA 02167, (617) 739-2000.

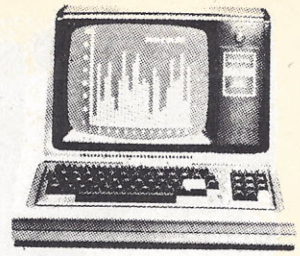
Dec 1-3 Computer Crime Info Conference, Crystal City Marriott Hotel, Arlington, VA, in-depth discussions of solutions to computer security issues for government and business executives. Information Exchange, 1730 N. Lynn St., Suite 400, Arlington, VA 22209, (703) 521-6209.

Dec 4 California Computer Shows, Hyatt-Palo Alto, Palo Alto, CA, OEM and end-user computer and peripheral products exhibited and demonstrated. Also held Apr 23 1981 at Hyatt and Mar 12 1981 at Inn-at-the-Park in Anaheim, CA. Norm De Nardi Enterprises, 95 Main St., Los Altos, CA 94022, (415) 941-8440.

Jan 27-29 Advanced Semiconductor Equipment Exposition, San Jose Convention Center, San Jose, CA, new products and emerging technology in semiconductor equipment explored in seminars and exhibits. Cartlidge & Assoc., 491 Macara Ave., Suite 1014, Sunnyvale, CA 94086, (408) 245-6870.

Feb 24-26 Nepcon West '81, Convention Center, Anaheim, CA, PCB/PWB/microelectronic materials, hardware, tools, supplies, and test instruments for engineering packaging/ production specialists. ISCM, 222 W. Adams St., Chicago, IL 60606, (312) 263-4866.

AUTHORIZED TRS-80® DEALER A301

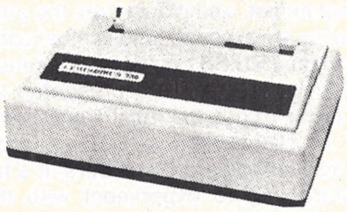


COMPUTER SPECIALISTS

10% DISCOUNT Off List
26-4002
64K 1 Drive
$3499.00

26-1056 16K Level II System with Keypad	$688.00
26-1145 RS-232 Board	84.00
26-1140 "O" K Interface	249.00
26-1141 "16" K Interface	365.00
26-1142 "32" K Interface	476.00
26-1160 Mini Disk - Drive O	424.00
26-1161 Mini Disk - Additional	424.00
26-1154 Lineprinter II	720.00
26-1156 Lineprinter III	1799.00
26-1180 Voice Synthesiser	339.00
26-1181 VOXBOX	145.00
26-1104 Factory Upper/Lower Case Modifaction Installed	70.00
26-1506 Scripsit - Tape	60.00
26-1563 Scripsit - Disk	85.00

15% DISCOUNT Off List
26-1054
4K Level II
$552.00



ALL OTHER R.S. SOFTWARE FURNITURE, STANDS, CABLES AND ACCESSORIES DEDUCT 10% FROM CATALOG PRICE

Model II Cobol Compiler
$360.00
Cobol Run Time Package
$36.00

CENTRONICS

Fast 100 CPS Centronics 730 Printer - $675.00
Text Quality Centronics 737 Printer - $850.00

Novation Cat Modem. . $149.00
CCA Data Management System 72.00
Adventure Games
Games 1-9 each 14.00

Acorn Software Products, Inc.

GAMES:

Alien Invasion	$9.00
Stock Market	9.00
Star Trek	9.00
Block 'Em	9.00
Ting-Tong	9.00

UTILITIES:

System Savers	14.00

EDUCATION:

Language Teacher	18.00

MICROSOFT

Model I Basic Compiler	$180.00
Model II Basic Compiler	360.00

BASF

10-5¼ Diskettes	$45.00
10-8" Diskettes	47.00

1-800-841-0860 Toll Free Order Entry

MICRO MANAGEMENT SYSTEMS, INC.

No Taxes on Out Of State Shipments

Immediate Shipment From Stock.

DOWNTOWN PLAZA SHOPPING CENTER
115 C SECOND AVE. S.W.
CAIRO, GEORGIA 31728
(912) 377-7120 Ga. Phone No.

Full Factory Warranty on All Items Sold.

Largest Inventory In the S.E. U.S.A.

BOOK REVIEWS

TRS-80 Basic (A Self-Teaching Guide)
By Bob Albrecht, Don Inman and Ramon Zamora
John Wiley & Sons, New York, NY

Reviewed by Tom Fox

If you're a rank beginner in computing with a TRS-80 I equipped with level II Basic, your novice status won't last long. TRS-80 Basic leads you by the hand through the maze of programming the computer to perform both entertaining and useful tasks.

The material begins at the simplest possible level and progresses steadily on to more complex concepts. None of it, however, is beyond the range of a bright junior high school student. The material is presented as hundreds of bite-size "frames," each explaining a single aspect of programming skill. Each frame is followed by a simple quiz intended to fix into your mind the material just presented, and each of the chapters closes with an in-depth examination. Over and over again, the reader is exhorted to "think like a computer" by tracing through a program and predicting how the machine will respond to each step. This is perhaps the most important yet least familiar of the skills required of a programmer, and it deserves the attention it gets. The book teaches by example, and is most useful when propped next to a warm TRS-80 computer. Dozens of programs—each with many variations— are presented and should actually be run in order to get the most from them.

Material is presented in slightly unconventional order. Strings, for example, are discussed before numbers. It's true that strings are more interesting to experiment with than numbers (particularly for math haters), but the authors saddle themselves with explaining this more complex subject first. Similarly, games are about all you will find in the early chapters, even though their appearance requires a description of the RND(X) random number generator—one of the more advanced Basic functions. It all seems to work, however, and the entertainment level is high.

Its greatest shortcoming is that it will be of little use to owners of other computers. The very complete treatment of the TRS-80 graphics capabilities, for example, is all but useless when related to another machine—even the TRS-80 II. □

351 pages $8.95

Dr. Dobb's Journal of Computer Calesthenics and Orthodontia—Volumes 2 & 3
Edited by Tom Williams
The People's Computer Co., Rochelle Park, NJ

Reviewed by Rocky Smolin

For the dyed-in-the-wool hobbyist who is just as happy working on a breadboard as a keyboard, these volumes are a compilation of almost exact reprints of 1977 and 1978 issues. For recent subscribers or those who failed to save their copies, they are just what the doctor ordered.

In addition to many short articles, mostly in favor of spiffy assembly language routines for micros, there are announcements of new products (many now obsolete), along with updates, corrections and enhancements. Much of this is dated, but still too new to have much nostalgia value. Names like SWTPC, Kim, Dazzle, Pilot, SC/MP, Stubal, and Sam76 appear frequently. As one not overly involved in assembly language routines or clever interface designs, I had limited interest.

The most enjoyable aspects are the full page graphic compositions by David Dameron. These are not only well executed technically, but also aesthetically pleasing, generated by what appears to be a fairly high resolution plotter.

There is one business article, a journal balancing program, well documented with complete source (in assembler). There is an excellent article telling everything you'd ever want to know about the universal product code. There are many sophisticated utility programs, compilers, disassemblers, linkage editors, relocators, an RTOS for the Intel MDS, macro generators, and operating systems, mostly well explained and documented with complete source listings. Of course, this means eye-straining hours of copying assembly language programs from oversized pages, some not reproduced with the sharpest clarity.

Strangely enough, these programs seem to be weighted rather heavily in favor of disassemblers: an 8080, an 8080 in MITS Basic, an M6800, an 8080 and support software for the SOL, a Z80 disassembler in TDL 8K Basic, and a compact 8080 and dumper. There are some entries on Pilot, Lisp and Forth, but little Basic and no Fortran or Pascal.

There are some good tutorials including a review of Forth for micros and an excellent article on list processing. □

Each volume: 479 pages $18.95

Microcomputer Primer
By Mitchell Waite and Michael Pardee
Howard W. Sams, Indianapolis, IN

Reviewed by Roger H. Edelson

This book, thankfully, improves as it goes on. By chapter 3, the authors hit their stride and begin to write a usable book. At this point and through to chapter 6, the descriptions of subsystems of a basic computer (CPU and power, memory, I/O, and programming) are generally well-written and error-free. A notable exception is oversimplification in the discussion of switching supplies.

The discussion of various CPUs and memory types is well done if one ignores the obvious bias in favor of Motorola's MPU architecture and dynamic memories. No mention is made of the problems associated with dynamic memories—finite soft bit error rate due to alpha particle events and the measures and costs needed to provide parity error correction schemes.

The section on programming provides good low-level coverage needed by newcomers. Attention has been paid to the various methods of addressing that may be implemented by a CPU. There is, however, no real discussion of the problems associated with keeping track of the program counter, and the stack is never covered.

In the chapter on operating systems, Basic is the only high-level language included. There is a good description of the differences between compilers and interpreters.

With the exception of those covering number systems and microcomputer math tables, the appendices could have been left out. The problem with tabulations of specific devices is that within six months, the list is usually out of date. The inclusion of pin-outs for selected ICs is definitely out of place, as no one could use this elementary book as a design aid.

The glossary is rather incomplete. For example, PL-1, a language probably unavailable to almost all microcomputer installations, is mentioned; Pascal is not. There are no references to top-down or structured programming, no stack, no 'fifo,' the list of omissions is almost endless—even including CP/M. □

384 pages $11.95

Place a special order
Continued from Page 62

LISTING 1

```
10  REM  MLS INDEX CREATE
20  REM  AUTHOR GARY A STOTTS
30 D$ =  CHR$ (4)
40  CALL  - 936
50  PRINT  TAB( 10);"CREATE MLS FILE": PRINT
60 N = 1144: REM   DIR SIZE = LOC  * PRICE * BEDROOMS
70  PRINT "ENTER NUMBER OF DISK DRIVES (1 - 3)
80  INPUT "FOR DATA STORAGE ";DN
90  IF DN < 1 OR DN > 3 THEN 70
100  PRINT D$;"OPEN MLS INDEX, S6, D1"
110  PRINT D$;"DELETE MLS INDEX"
120  PRINT D$;"OPEN MLS INDEX"
130  PRINT D$;"WRITE MLS INDEX"
140  PRINT N: PRINT O: PRINT DN: PRINT "": REM  DIR SIZE, NEXT REC ADDR, # DISKS
150  FOR C = 1 TO N
160  PRINT O: PRINT O: NEXT : REM  INDEX, POINTER
170  PRINT D$;"CLOSE"
180  END
```

LISTING 2

```
10  REM  MLS SYSTEM
20  REM  AUTHOR GARY A STOTTS
30  ONERR  GOTO 3030
40  CALL  - 936: INPUT "DO YOU WANT PRINTED OUTPUT Y OR N ";PR$
50 D$ =  CHR$ (13) +  CHR$ (4): DIM DT$(10): DIM GT$(10)
60  PRINT D$;"OPEN MLS DATA 1, L64, S6, D1"
70  PRINT D$;"OPEN MLS INDEX, S6, D1"
80  PRINT D$;"READ MLS INDEX"
90  INPUT DS: INPUT NXT: INPUT ND: INPUT FS$
100  DIM DI%(DS): DIM DP%(DS)
110  FOR C = 1 TO DS
120  INPUT DI%(C): INPUT DP%(C): NEXT
130  PRINT D$;"CLOSE MLS INDEX"
140  IF ND >  = 2 THEN  PRINT D$;"OPEN MLS DATA 2, L64, S6, D2"
150  IF ND >  = 3 THEN  PRINT D$;"OPEN MLS DATA 3,L64, S7,D1"
160  REM
170  GOSUB 2530: REM  FILL ARRAYS
180  CALL  - 936
189.6  PRINT  TAB( 10);"MLS PROGRAM MENU": PRINT
200  PRINT "1 - LOOK-UP MLS PROPERTY"
210  PRINT "2 - ADD A MLS PROPERTY"
220  PRINT "3 - SELL A MLS PROPERTY"
230  PRINT "4 - STOP"
240  PRINT "5 - REORGANIZE"
250  PRINT : INPUT "    ENTER YOUR SELECTION ";A$:SL =  VAL (A$)
260  IF SL < 1 OR SL > 5 THEN 180
270  ON SL GOSUB 320,450,650,880,2710
280  IF SL <  > 4 THEN 180
290  END
300  REM
310  REM  LOOK-UP
320  GOSUB 1370
330 F1 =  - 1: GOSUB 1210
340  IF R = 0 THEN  PRINT "PROPERTY NOT FOUND": PRINT "HIT RETURN TO CONTINUE":
     CALL  - 756: RETURN
350 R = DP%(R): IF PR$ = "Y" THEN  PRINT D$;"PR#1"
360  PRINT D$;"READ MLS DATA ";WD;",R";R - DF
370  FOR C = 1 TO 10: INPUT DT$(C): NEXT : PRINT D$
380  GOSUB 990: REM  DISPLAY
390 R =  VAL (DT$(10))
400  IF R > 0 THEN  GOSUB 1310: GOTO 360
410  PRINT D$;"PR#0"
420  RETURN
430  REM
440  REM  ADD PROPERTY
450  GOSUB 1370
460  GOSUB 2070
470 F1 = 1: GOSUB 1210
480  IF R = 0 THEN  PRINT "DIRECTORY FULL - REORGANIZE":
     PRINT "HIT RETURN TO CONTINUE": CALL  - 756: RETURN
490  IF  LEN (FS$) = 0 THEN NXT = NXT + 1:PP = NXT: GOTO 510
500  GOSUB 580
510 LNK = 0: IF DI%(R) = I1 THEN LNK = DP%(R): REM  CHAIN
1370  CALL  - 936
1380  PRINT  TAB( 10);"INPUT ATTRIBUTES": PRINT
1390  PRINT  TAB( 14);"LOCATION": PRINT
1400  PRINT " 1 - DENVER SW"
1410  PRINT " 2 - DENVER SE"
1420  PRINT " 3 - DENVER NE"
1430  PRINT " 4 - DENVER NW"
1440  PRINT " 5 - NORTH SUBURBAN WEST"
1450  PRINT " 6 - NORTH SUBURBAN CENTRAL"
1460  PRINT " 7 - NORTH SUBURBAN EAST"
1470  PRINT " 8 - SOUTH SUBURBAN WEST"
1480  PRINT " 9 - SOUTH SUBURBAN CENTRAL"
1490  PRINT "10 - SOUTH SUBURBAN EAST"
1500  PRINT : INPUT "    ENTER LOCATION OR HIT ENTER ";A$
1510 LOC =  VAL (A$): IF LOC > 10 THEN 1370
1520  IF LOC > 0 THEN 1780
1530  HTAB (1): VTAB (5): CALL  - 958
1540  PRINT "11 - AURORA NORTH"
1550  PRINT "12 - AURORA CENTRAL"
1560  PRINT "13 - AURORA SOUTH
1570  PRINT "14 - MONTBELLO"
1580  PRINT "15 - BROOMFIELD"
1590  PRINT "16 - JEFFCO NORTH"
1600  PRINT "17 - JEFFCO WEST"
1610  PRINT "18 - JEFFCO CENTRAL"
1620  PRINT "19 - JEFFCO SOUTH"
1630  PRINT : INPUT "    ENTER LOCATION OR HIT RETURN ";A$
1640 LOC =  VAL (A$): IF (LOC < 11 OR LOC > 19) AND LOC <  > 0 THEN 1530
1650  IF LOC > 0 THEN 1780
1660  VTAB (5): HTAB (1): CALL  - 958
1670  PRINT "20 - MOUNTAIN AREA 1"
1680  PRINT "21 - MOUNTAIN AREA 2"
1690  PRINT "22 - MOUNTAIN AREA 3"
1700  PRINT "23 - MOUNTAIN AREA 4"
1710  PRINT "24 - MOUNTAIN AREA 5"
1720  PRINT "25 - MOUNTAIN AREA 6"
1730  PRINT "26 - OTHER"
1740  PRINT : INPUT "    ENTER LOCATION ";A$
1750 LOC =  VAL (A$): IF LOC = 0 THEN 1370
1760  IF (LOC < 20 OR LOC > 26) THEN 1660
1770  REM
1780  VTAB (3): HTAB (1): CALL  - 958
1790  PRINT  TAB( 12);"ASKING PRICE": PRINT
1800  PRINT " 1 - 00,000 - 29,999"
1810  PRINT " 2 - 30,000 - 39,999"
1820  PRINT " 3 - 40,000 - 49,999"
1830  PRINT " 4 - 50,000 - 59,999"
1840  PRINT " 5 - 60,000 - 69,999"
1850  PRINT " 6 - 70,000 - 79,999"
1860  PRINT " 7 - 80,000 - 89,999"
1870  PRINT " 8 - 90,000 - 99,999"
1880  PRINT " 9 - 100,000 - 119,999"
1890  PRINT "10 - 120,000 - 139,999"
1900  PRINT "11 - MORE THAN 140,000"
1910  PRINT : INPUT "ENTER ASKING PRICE ";A$
1920 CST =  VAL (A$)
1930  IF CST < 1 OR CST > 11 THEN 1780
1940  REM
1950  VTAB (3): HTAB (1): CALL  - 958
1960  PRINT  TAB( 10);"HOW MANY BEDROOMS": PRINT
1970  PRINT "1 - 1 BEDROOM"
1980  PRINT "2 - 2 BEDROOMS"
1990  PRINT "3 - 3 BEDROOMS"
2000  PRINT "4 - 4 OR MORE BEDROOMS"
2010  PRINT : INPUT "ENTER # BEDROOMS ";A$
2020 BR =  VAL (A$)
2030  IF BR < 1 OR BR > 4 THEN 1950
2040  RETURN
2050  REM
2060  REM  INPUT DATA
2070  CALL  - 936
2080  PRINT  TAB( 10);"MLS PROPERTY DETAILS": PRINT
2090  INPUT "MLS NUMBER (N4) ";A$:MLS =  VAL (A$)
2100  IF MLS < 1 OR MLS > 9999 THEN 2090
2110  INPUT "ADDRESS (A20) ";ADR$
2120  IF  LEN (ADR$) > 20 THEN 2110
2130  INPUT "ASKING PRICE (N6) ";PR$
2140  IF  LEN (PR$) > 6 OR  VAL (PR$) < 1 THEN 2130
2150  INPUT "# BEDROOMS (N1) ";BR$
2160  IF  LEN (BR$) > 1 THEN 2150
2170  INPUT "# BATHS (N3.2) ";BTH$
2180  IF  LEN (BTH$) > 4 THEN 2170
2190  INPUT "SQUARE FEET (N4) ";SF$
2200  IF  LEN (SF$) > 4 THEN 2190
```

```
520 DI%(R) = I1:DP%(R) = PP
530  PRINT D$;"WRITE MLS DATA ";WD;",R";PP - DF
540  PRINT "A": PRINT MLS: PRINT ADR$: PRINT PR$
550  PRINT BR$: PRINT BTH$: PRINT TRM$: PRINT SF$
560  PRINT SCH$: PRINT LNK: PRINT D$
570  RETURN
580 PP =  VAL ( LEFT$ (FS$,4))
590  IF  LEN (FS$) = 4 THEN FS$ = ""
600  IF  LEN (FS$) > 4 THEN FS$ =  RIGHT$ (FS$, LEN (FS$) - 4)
610 X =  FRE (0)
620  RETURN
630  REM
640  REM  SOLD
650  GOSUB 1370
660  CALL  - 936
670  INPUT "ENTER MLS NUMBER ";A$
680  IF  LEN (A$) > 4 THEN 660
690 F1 =  - 1: GOSUB 1210
700  IF R = 0 THEN  PRINT "PROPERTY NOT FOUND": PRINT "HIT RETURN TO CONTINUE": CALL  - 756: RETURN
710 R = DP%(R)
720  PRINT D$;"READ MLS DATA ";WD;",R";R - DF
730  FOR C = 1 TO 10: INPUT DT$(C): NEXT : PRINT D$
740  IF DT$(2) = A$ THEN 790
750 R =  VAL (DT$(10))
760  IF R = 0 THEN  RETURN
770  GOSUB 1310: GOTO 720
780  REM  SELL IT
790  PRINT "ENTER E FOR EARNEST MONEY OR S FOR"
800  INPUT "SOLD OR D FOR DELET ";A$
810  IF A$ <  > "E" AND A$ <  > "S" AND A$ <  > "D" THEN 790
820 DT$(1) = A$
830  PRINT D$;"WRITE MLS DATA ";WD;",R";R - DF
840  FOR C = 1 TO 10: PRINT DT$(C): NEXT : PRINT D$
850  RETURN
860  REM
870  REM  END PROGRAM
880  PRINT D$;"CLOSE"
890  PRINT D$;"DELETE MLS INDEX"
900  PRINT D$;"OPEN MLS INDEX, S6, D1"
910  PRINT D$;"WRITE MLS INDEX"
920  PRINT DS: PRINT NXT: PRINT ND: PRINT FS$
930  FOR C = 1 TO DS
940  PRINT DI%(C): PRINT DP%(C): NEXT
950  PRINT D$;"CLOSE"
960  RETURN
970  REM
980  REM  PRINT MLS LIST
990  CALL  - 936
1000  PRINT  TAB( 8);"MLS PROPERTY ";DT$(2): PRINT
1010  FOR C = 1 TO 4: IF DT$(1) =  LEFT$ (ST$(C),1) THEN 1040
1020  NEXT
1030 C = 1
1040  PRINT "PRICE ";DT$(4); SPC( 3);"SALES STATUS "; RIGHT$ (ST$(C), LEN (ST$(C)) - 1)
1050  PRINT "ADDRESS ";DT$(3)
1060  PRINT "# BEDROOMS ";DT$(5);" AND BATHS ";DT$(6)
1070  PRINT "TERMS ";
1080  FOR C = 1 TO 6
1090  IF  MID$ (DT$(7),C,1) = "Y" THEN  PRINT T$(C);" ";
1100  NEXT
1110  PRINT "  SQ FT ";DT$(8)
1120  PRINT "STYLE ";S$( VAL ( LEFT$ (DT$(9),1)))
1130  PRINT "CONSTRUCTION ";C$( VAL ( MID$ (DT$(9),2,1)))
1140  PRINT "HEATING METHOD ";H$( VAL ( RIGHT$ (DT$(9),1)))
1150  PRINT "LINK ";DT$(10)
1160  IF PR$ = "Y" THEN  RETURN
1170  PRINT : PRINT "HIT RETURN TO CONTINUE": CALL  - 756
1180  RETURN
1190  REM
1200  REM  HASH
1210 I1 = 6 * BR + 3 * CST + LOC - 10:TRY = 0
1220 R = BR * CST * LOC
1230 TRY = TRY + 1: IF TRY > DS OR R > DS THEN R = 0: RETURN
1240 WD = 1:DF = 0
1250  IF DP%(R) > 1000 AND DP%(R) <  = 2400 THEN WD = 2:DF = 1000
1260  IF DP%(R) > 2400 AND DP%(R) <  = 3800 THEN WD = 3:DF = 2400
1270  IF DI%(R) = 0 AND F1 > 0 THEN  RETURN : REM  ADDITION
1280  IF DI%(R) = I1 THEN  RETURN
1290 R = R + 1: GOTO 1230
1300  REM
1310 WD = 1:DF = 0
1320  IF R > 1000 AND R <  = 2400 THEN WD = 2:DF = 1000
1330  IF R > 2400 AND R <  = 3800 THEN WD = 3:DF = 2400
1340  RETURN
1350  REM
1360  REM
2210  VTAB (3): HTAB (1): CALL  - 958:TRM$ = ""
2220  PRINT  TAB( 10);"TERMS"
2230  FOR C = 1 TO 6
2240  PRINT C;" - ";T$(C);
2250  INPUT "   ENTER Y OR N ";A$
2260  IF A$ <  > "N" AND A$ <  > "Y" THEN 2250
2270 TRM$ = TRM$ + A$
2280  NEXT
2290  VTAB (3): HTAB (1): CALL  - 958
2300  PRINT  TAB( 10);"STYLE": PRINT
2310  FOR C = 1 TO 9
2320  PRINT C;" - ";S$(C): NEXT
2330  PRINT : INPUT "ENTER STYLE ";A$
2340  IF  LEN (A$) > 1 OR  VAL (A$) = 0 THEN 2330
2350 SCH$ = A$
2360  VTAB (3): HTAB (1): CALL  - 958
2370  PRINT  TAB( 10);"CONSTRUCTION": PRINT
2380  FOR C = 1 TO 9
2390  PRINT C;" - ";C$(C): NEXT
2400  PRINT : INPUT "ENTER CONSTRUCTION ";A$
2410  IF  LEN (A$) > 1 OR  VAL (A$) = 0 THEN 2400
2420 SCH$ = SCH$ + A$
2430  VTAB (3): HTAB (1): CALL  - 958
2440  PRINT  TAB( 10);"HEATING METHOD": PRINT
2450  FOR C = 1 TO 8
2460  PRINT C;" - ";H$(C): NEXT
2470  PRINT : INPUT "ENTER HEATING METHOD ";A$
2480  IF  LEN (A$) > 1 OR  VAL (A$) = 0 THEN 2470
2490 SCH$ = SCH$ + A$
2500  RETURN
2510  REM
2520  REM
2530  DIM ST$(4): DIM S$(9): DIM C$(9): DIM H$(9): DIM T$(6)
2540  DATA  AFOR SALE,DDELETED,EEARNEST MONEY,SSOLD
2550  FOR C = 1 TO 4: READ ST$(C): NEXT
2560  DATA  CASH,CONV,ASSUME,FHA,VA,OWC
2570  FOR C = 1 TO 6: READ T$(C): NEXT
2580  DATA  RANCH,BI-LEVEL,TRI-LEVEL,TWO-STORY,BUNGALOW
2590  DATA  CHALET,MOBILE HOME,A-FRAME,OTHER
2600  FOR C = 1 TO 9: READ S$(C): NEXT
2610  DATA  FRAME,BRICK,FRAME/BRICK,STUCCO,MOSS ROCK
2620  DATA  ADOBE,LOG,MODULAR,OTHER
2630  FOR C = 1 TO 9: READ C$(C): NEXT
2640  DATA  GAS,ELECTRIC,PROPANE,SOLAR
2650  DATA  WOOD,COAL,OIL,OTHER
2660  FOR C = 1 TO 8: READ H$(C): NEXT
2670  RETURN
2680  REM
2690  REM
2700  REM  REORG
2710  FOR C = 1 TO DS
2720  IF DP%(C) <  > 0 THEN  GOSUB 2760
2730  NEXT
2740  PRINT D$: RETURN
2750  REM
2760 R = DP%(C):DT$ = "N":RS = 0:X =  FRE (0)
2770  GOSUB 1310
2780  PRINT D$;"READ MLS DATA ";WD;",R";R - DF
2790  FOR D = 1 TO 10: INPUT DT$(D): NEXT
2800  IF DT$(1) = "D" OR DT$(1) = "S" THEN  GOSUB 2880: GOTO 2830
2810  IF DT$ = "Y" THEN  GOSUB 2930
2820 RS = R
2830 R =  VAL (DT$(10))
2840  IF R > 0 THEN 2770
2850  IF DT$ = "Y" THEN  GOSUB 2930
2860  RETURN
2870  REM
2880 A$ =  STR$ (R):DT$ = "Y": REM  FREE STORAGE LIST
2890  IF  LEN (A$) < 4 THEN A$ = "0" + A$: GOTO 2890
2900 FS$ = FS$ + A$
2910  RETURN
2920  REM  UPDATE LAST ACTIVE
2930 DT$ = "N": IF RS = 0 AND R = 0 THEN DI%(C) = 0:DP%(C) = 0: RETURN : REM  ALL DELETED
2940  IF RS = 0 AND R > 0 THEN DP%(C) = R: RETURN : REM  NEW 1ST REC
2950 SV = R:R = RS: GOSUB 1310
2960  PRINT D$;"READ MLS DATA ";WD;",R";R - DF
2970  FOR D = 1 TO 10: INPUT GT$(D): NEXT
2980 GT$(10) =  STR$ (SV)
2990  PRINT D$;"WRITE MLS DATA ";WD;",R";R - DF
3000  FOR D = 1 TO 10: PRINT GT$(D): NEXT
3010 R = SV
3020  RETURN
3030  PRINT "ERROR NO "; PEEK (222);" AT LINE "; PEEK (218) +  PEEK (219) * 256:
      PRINT D$;"CLOSE": END
```

Binary Cents

Continued from Page 72

PROGRAM LISTING

```
00010 '           ***************************
00020 '           ***   SMART CHECKBOOK   ***
00030 '           ***************************
00040 '
00050 '     Written in Microsoft BASIC-80, BASCOM 5.01
00060 '     Written by M. L. Lesser, 11-25-79
00070 '
00080 '     This program finds a subset of the outstanding
00090 '     checks that corresponds to the amount paid by
00100 '     the bank between two readings of the bank's
00110 '     record of your balance.
00120 '
00200 '
00210 '     The following integer variables are used:
00220 '
00230 '        I,IX,J:      Indices
00240 '        CHECKNO(I):  The number of the Ith outstanding check
00250 '        STACK(J):    List of checks paid
00260 '
00270 '     The following variables are carried in double-precision to
00280 '        maintain accuracy to the penny if any total exceeds $9999.99:
00290 '
00300 '        AMOUNT(I):   Amount of CHECKNO(I)
00310 '        OBANBAL:     Bank's balance at previous reading
00320 '        NBANBAL:     Bank's balance at this reading
00330 '        OMYBAL:      Checkbook balance at previous reading
00340 '        MYBAL:       Checkbook balance at this reading
00350 '        VALUE:       Value of all checks paid during period
00360 '        REMAINDER:   Working storage
00370 '        TEMP:        Working storage
00380 '        WORKING:     Working storage
00390 '
00400 '     The following are string variables:
00410 '
00420 '        DATE:        Today's date
00430 '        PDATE:       Date of previous reading
00440 '        PAY(I):      Payee of CHECKNO(I)
00450 '        DUMMY:       Working storage
00460 '
00470 '
00500 '     Setup
00510 '
00520 DEFINT C,I,J,S
00530 DEFDBL A,M,N,O,R,T,V,W
00540 DEFSTR D,P
00550 DIM AMOUNT(20), CHECKNO(20), PAY(20), STACK(20)
00560 DEF FNA(W) = INT(100*W+0.5)            'Penny function
00570 '
00580 PRINT CHR$(27) "K" STRING$(20,0)       'Clear screen
00590 '
00600 '     Main program
00610 '
00620 LINE INPUT "Enter today's date (MM-DD-YY):  "; DATE
00630 GOSUB 1000                             'Enter checks
00640 GOSUB 2000                             'Enter deposits
00650 GOSUB 3000                             'Enter new balances
02100 LPRINT "Total deposits since " PDATE ":  ";
02110 LPRINT USING "$$####.##     "; REMAINDER
02120 LPRINT
02130 RETURN
02140 '
02150 '
03000 '       Enter new balances
03010 PRINT
03020 INPUT "Current checkbook balance"; DUMMY
03030 IF (LEFT$(DUMMY,1) = "S" OR LEFT$(DUMMY,1) = "s")
          THEN MYBAL = OMYBAL ELSE MYBAL = VAL(DUMMY)
03040 PRINT
03050 INPUT "New bank balance"; DUMMY
03060 IF (LEFT$(DUMMY,1) = "S" OR LEFT$(DUMMY,1) = "s")
          THEN NBANBAL = OBANBAL ELSE NBANBAL = VAL(DUMMY)
03070 LPRINT "New bank balance on " DATE " is";
03080 LPRINT USING "$$####.##     "; NBANBAL
03090 LET VALUE = OBANBAL + REMAINDER - NBANBAL
03100 LET REMAINDER = TEMP
03110 LPRINT "Total value of checks paid since " PDATE " is";
03120 LPRINT USING "$$####.##     "; VALUE
03130 LPRINT "Value of checks still outstanding on " DATE " is";
03140 LPRINT USING "$$####.##     "; REMAINDFR - VALUE
03150 IF FNA(VALUE) < 0 GOTO 7500               'Input error
03160 LET TEMP = REMAINDER - VALUE
03170 IF FNA(TEMP) < 0 GOTO 7500                'Input error
03180 LPRINT "Checkbook balance on " DATE " is";
03190 LPRINT USING "$$####.##     "; MYBAL
03200 LET WORKING = NBANBAL - MYBAL
03210 IF FNA(TEMP) <> FNA(WORKING) GOTO 7500  'Input error
03220 RETURN
03230 '
04000 '       Search
04010 LET TEMP = VALUE: I = 0: STACK(0) = 0
04020 WHILE FNA(TEMP) <= FNA(REMAINDER) AND FNA(TEMP) <> 0
04030       LET I = I + 1
04040       LET REMAINDER = REMAINDER - AMOUNT(I)
04050       IF FNA(AMOUNT(I)) > FNA(TEMP) GOTO 4090
04060       LET TEMP = TEMP - AMOUNT(I)
04070       LET J = STACK(0) + 1: STACK(0) = J
04080       LET STACK(J) = I
04090 WEND
04100 '
04110 '       Done?
04120 IF FNA(TEMP) = 0 THEN RETURN
04130 '
04140 '       Not done--back up one in stack
04150 LET J = STACK(0): STACK(0) = J - 1
04160 IF STACK(0) < 0 GOTO 8000               'No solution found
04170 LET I = STACK(J): REMAINDER = 0
04180 FOR IX = I + 1 TO CHECKNO(0)
04190       LET REMAINDER = REMAINDER + AMOUNT(IX)
04200 NEXT IX
04210 LET TEMP = TEMP + AMOUNT(I)
04220 PRINT USING "####     "; CHECKNO(I);
04230 GOTO 4020
04240 '
04250 '
05000 '       Print and file results
05010 LET TEMP = 0
05020 LPRINT: LPRINT "Checks paid since " PDATE ":"
```

```
00660   GOSUB 4000                               'Search
00670   GOSUB 5000                               'Results
00680   '
00690   END
01000   '       Enter checks - read old file
01010   LPRINT "This run made on " DATE: LPRINT
01020   ON ERROR GOTO 1800
01030   LET I = 0: TEMP = 0
01040   OPEN "I",1,"CHECKS"
01050   INPUT#1, PDATE, OBANBAL, WORKING, OMYBAL
01060   LPRINT "On " PDATE " bank balance was ";
01070   LPRINT USING "$$####.##     "; OBANBAL
01080   LPRINT "Value of checks outstanding was ";
01090   LPRINT USING "$$####.##     "; WORKING
01100   LPRINT
01110   LPRINT "Outstanding checks on " PDATE " were:"
01120   WHILE NOT EOF(1)
01130           LET I = I + 1
01140           INPUT#1, CHECKNO(I), AMOUNT(I), PAY(I)
01150           GOSUB 6000                       'Print check data
01160   WEND
01170   IF FNA(TEMP) <> FNA(WORKING) GOTO 7000   'File data error
01180   LPRINT
01190   '
01200   '       Enter new checks
01210   LPRINT "Checks written since " PDATE ":"
01220   PRINT "Enter new checks"
01230   CLOSE
01240   LET I = I + 1
01250           INPUT "     CHECKNO (Enter '0' to quit)"; CHECKNO(I)
01260           IF CHECKNO(I) = 0 GOTO 1310
01270           INPUT "     AMOUNT"; AMOUNT(I)
01280           LINE INPUT "     PAYEE ?"; PAY(I)
01290           GOSUB 6000                       'Print check data
01300   GOTO 1240
01310   LET CHECKNO(0) = I - 1
01320   LPRINT "Total value of checks outstanding: ";
01330   LPRINT USING "$$####.##     "; TEMP
01340   RETURN
01350   '
01800   IF ERR = 53 AND ERL = 1040 GOTO 1900
01810   ON ERROR GOTO 0
01900   PRINT "NO PREVIOUS FILE"
01910   '       Missing file data
01920   INPUT "Old bank balance"; OBANBAL
01930   INPUT "Date of old bank balance"; PDATE
01940   LPRINT "On " PDATE " bank balance was ";
01950   LPRINT USING "$$####.##     "; OBANBAL
01960   LPRINT "Checks outstanding are:"
01970   GOTO 1220
01980   '
02000   '       Enter deposits
02010   PRINT: PRINT "Enter deposits separately"
02020   LPRINT
02030   LPRINT "Deposits made since " PDATE ":"
02040   LET REMAINDER = 0
02050   INPUT "     Deposit (enter '0' to quit)"; WORKING
02060           IF WORKING = 0 GOTO 2100
02070           LPRINT,: LPRINT USING "$$####.##     "; WORKING
02080           LET REMAINDER = REMAINDER + WORKING
02090   GO TO 2050
05030   FOR J = 1 TO STACK(0)
05040           LET I = STACK(J)
05050           GOSUB 6000                       'Print check data
05060           LET CHECKNO(I) = 0
05070   NEXT J
05080   LPRINT "Total value of checks paid: ";
05090   LPRINT USING "$$####.##     "; TEMP
05100   IF FNA(TEMP) <> FNA(VALUE) GOTO 9000     'Error stop
05110   LET TEMP = 0
05120   LPRINT: LPRINT "Checks outstanding on " DATE ":"
05130   FOR I = 1 TO CHECKNO(0)
05140           IF CHECKNO(I) <> 0 THEN GOSUB 6000
05150   NEXT I
05160   LPRINT "Total value of checks outstanding: ";
05170   LPRINT USING "$$####.##     "; TEMP
05180   IF FNA(TEMP) <> FNA(WORKING) GOTO 9000   'Error stop
05190   '
05200   '       Store file data
05210   OPEN "O",1,"CHECKS"
05220   WRITE#1, DATE, NBANBAL, TEMP, MYBAL
05230   FOR I = 1 TO CHECKNO(0)
05240           IF CHECKNO(I) = 0 GOTO 5260
05250           WRITE#1, CHECKNO(I), AMOUNT(I), PAY(I)
05260   NEXT I
05270   RESET
05280   RETURN
05290   '
06000   '       Print and sum check data
06010   LPRINT,
06020   LPRINT USING "####     "; CHECKNO(I);
06030   LPRINT USING "$$####.##     "; AMOUNT(I);
06040   LPRINT PAY(I)
06050   LET TEMP = TEMP + AMOUNT(I)
06060   RETURN
06070   '
06080   '
07000   '       File data error
07010   PRINT "FILE DATA DOES NOT BALANCE"
07020   PRINT "RE-ENTER DATA FROM CONSOLE"
07030   CLOSE
07040   GOTO 1910                                'No file data
07050   '
07060   '
07500   '       Input error
07510   PRINT "INPUT VALUES IN ERROR"
07520   PRINT "PROGRAM ABORTED"
07530   STOP
07540   '
07550   '
08000   '       No solution
08010   PRINT "NO SOLUTION FOUND"
08020   PRINT "PROGRAM ABORTED"
08030   END
08040   '
08050   '
09000   '       Error stop
09010   PRINT "SOLUTION DOES NOT BALANCE"
09020   PRINT "PROGRAM ABORTED"
09030   END
09040   '
09050   '
```

Program your phone calls

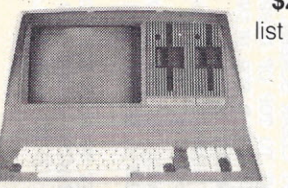
Continued from Page 78

PROGRAM LISTING

```
100 REM * TELEPHONE DIALER PROGRAM *
110 REM * COPYRIGHT 1979 FRED BLECHMAN * VERSION 8/28/79
115 REM * FAST DIAL (LINES 590,1020 & 1030)
116 REM * CHARGE CALCULATION (LINES 950-990)
117 REM * TIME COUNTING - LINE 780 *
120 CLS:PRINT:PRINTTAB(18)"TELEPHONE DIALER PROGRAM"
130 PRINT:INPUT"DO YOU WANT INSTRUCTIONS? Y/N";I$
140 IF LEFT$(I$,1)="Y" OR LEFT$(I$,1)="N" GOTO 150 ELSE 130
150 IF LEFT$(I$,1)="Y" GOSUB 9000
160 CLS:PRINT:PRINTTAB(20)"TELEPHONE DIALER PROGRAM"
170 PRINT:INPUT"LIST OF ALL NAMES? Y/N",B$
180 IF LEFT$(B$,1)="Y" OR LEFT$(B$,1)="N" GOTO 190 ELSE 170
190 IF LEFT$(B$,1)="Y" GOSUB 8000
200 REM * 14 TOTAL DIGITS MAY BE DIALED *
210 DIMY(14)
220 R=0
230 PRINT:PRINT
240 PRINT"          WHO DO YOU WANT TO DIAL.....??"
250 PRINT"ENTER: NAME OR R(=REDIAL) OR L(=LIST) OR M(=MANUAL)..."
260 INPUT"(NOTE: PHONE MUST BE OFF HOOK FOR DIALING....) ";A$
270 IF A$="R" GOTO 340
280 IF A$="M" GOTO 900
290 IF A$="L" GOSUB 8000 : GOTO 230
300 READ N$,A$,B$
310 IF N$="END" PRINTTAB(13)"NOT ON LIST! TRY AGAIN...":RESTORE:GOTO 230
320 IF N$=A$ GOTO 340
330 GOTO 300
340 PRINTTAB(8)"NAME";TAB(24)"AREA CODE";TAB(42)"NUMBER"
350 PRINTTAB(5)A$;TAB(25)B$;TAB(40)A$
360 REM * SEPARATE DIGITS, BACKWARDS, AND HOLD IN ARRAY *
370 A=INT(A$)
380 R=R+1
390 X=INT(A/10)
400 Y=INT(10*(A/10-X)+.5)
410 Y(R)=Y
420 IF X=0 GOTO 460
430 A=X
440 GOTO 380
450 REM * SAME WITH AREA CODE DIGITS *
460 IF B$=0 GOTO 560
470 B=INT(B$)
480 R=R+1
490 X=INT(B/10)
500 Y=INT(10*(B/10-X)+.5)
510 Y(R)=Y
520 IF X=0 GOTO 560
530 B=X
540 GOTO 480
550 REM * DIALS NUMBERS IN REVERSE (FORWARD) ORDER *
560 PRINT:N=R:FOR D=1 TO R
570 PRINTTAB(15)Y(N);:IF Y(N)=0 THEN Y(N)=10
580 GOSUB 1010
590 FOR T=1 TO 50:NEXTT
600 N=N-1:IF N=0 GOTO 620
610 NEXT D
620 RESTORE:PRINT
700 REM * TIMER ROUTINE *
710 PRINT"WHEN THE PARTY AT THE OTHER END PICKS UP THE RECEIVER, PRESS"
720 INPUT"ENTER TO START TIMING...OR ENTER 'D' TO DIAL AGAIN...";T$
730 IF T$="D" THEN T$=" ":GOTO 220
740 PRINT:PRINT"PRESS 'S' TO STOP...."
750 P=0:S=60:M=1
760 PRINT:PRINT"MINUTES CHARGED SO FAR...";M
770 PRINT"... SECONDS TO NEXT CHARGE...";S;
780 FOR X=1 TO 266:NEXT X
790 IF INKEY$="S" GOTO 950
800 S=S-1:IF S=0 GOTO 820
810 PRINT S;:GOTO 780
820 M=M+1:S=60:GOTO 760
900 REM * HAND DIALING *
910 PRINT:INPUT"ENTER LOCAL NUMBER (NO DASHES)";A$
920 INPUT"PICK UP PHONE. ENTER '1' IF REQUIRED & AREA CODE, OR 0...";B$
930 A$="DIAL":GOTO 340
950 PRINT:PRINT:INPUT"CHARGES? Y/N";T$
955 IF T$="N" GOTO 220
960 PRINT:PRINT"FIND UNASSISTED RATES IN PHONE BOOK FOR NUMBER CALLED....."
965 PRINT".....FOR THIS DAY AND HOUR!"
970 INPUT"ENTER FIRST MINUTE, EACH ADDED MINUTE (CENTS)=";AA,BB
975 CC=(AA/100)+(M-1)*(BB/100)
980 PRINT:PRINT"TOTAL CHARGE TO ";B$;"-";A$;" FOR ";M;" MINUTES IS $";CC
990 GOTO220
1000 REM * OPERATES TRS-80 RELAY FOR DIALING *
1010 FOR C=1 TO Y(N)
1020 OUT255,4:FOR T=1 TO 10:NEXT T
1030 OUT255,0:FOR T=1 TO 10:NEXT T
1040 NEXT C
1050 RETURN
2000 REM * NAMES AND NUMBERS STORED IN DATA STATEMENTS *
2010 DATA ADA ORDER,2534611,1800
2020 DATA ARROYO,8858102,0
2030 DATA BABB,3403818,0

2650 DATA WOODBRIDGE,9921370,0
2660 DATA WOLLTER,7478015,1714
7990 DATA END,0,0
8000 REM * PRINT LIST OF NAMES WHEN REQUESTED *
8010 PRINT:PRINTTAB(13)"PRESS SHIFT@ TO STOP LISTING..."
8020 PRINT:FOR T=1 TO 250:NEXT T
8030 READ N$,A,B:PRINT N$,
8040 IF N$="END" RESTORE:GOTO 8060
8050 GOTO 8030
8060 RETURN
9000 REM * INSTRUCTIONS WHEN REQUESTED *
9010 CLS
9020 PRINT"     ENTER YOUR OWN DATA STATEMENTS ALPHABETICALLY"
9030 PRINT"BEGINNING AT LINE 2010. SEE THE LISTING FOR SAMPLES."
9040 PRINT"        THE DATA ENTRIES ARE IN THIS FORMAT:"
9050 PRINT"       DATA NAME,LOCAL PHONE NUMBER,AREA CODE"
9060 PRINT"              (DO NOT USE DASHES!)"
9070 PRINT:PRINT"IMPORTANT!! IF THE AREA CODE IS THE SAME AS YOUR OWN CODE,"
9080 PRINT"THEN ENTER A ZERO. IF THE AREA CODE IS NOT THE SAME AS YOURS,"
9090 PRINT"AND YOU NEED TO ENTER A '1' BEFORE THE AREA CODE WHEN"
9100 PRINT"YOU DIAL NORMALLY, THEN ENTER A '1' BEFORE THE AREA CODE"
9110 PRINT"IN THE DATA STATEMENT!"
9120 PRINT:PRINT"     SEE THE ACCOMPANYING SCHEMATIC FOR INSTRUCTIONS"
9130 PRINT"ON HOW TO CONNECT YOUR TRS-80 TO THE PHONE LINE...."
9140 PRINT:INPUT"WHEN READY TO DIAL, PRESS ENTER....";D$
9150 RETURN
```

SUPERBRAIN

FUTRA PRICE
$2445
list $2995



WOW! A complete expandable, Z80, S100, CP/M, Computer System. With Dual Drives 350K, 12" CRT, 32K RAM Double Density Superbrain $$ CALL

ATARI 800

FUTRA PRICE
$795
list $1085

A 16K RAM personal computing system with BASIC Lang.
Program Recorder $78
Education Cartridge $58

Apple II or Apple II Plus w/16K RAM

List–$1195.00
FUTRA Price–$939.95
(ship. included)

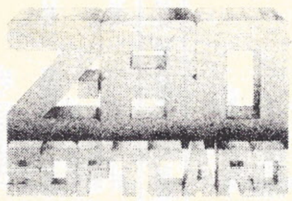
Disk II with controller	$498.00
Disk II 2nd drive	449.00
Graphics Tablet	679.00
Language System with PASCAL	397.00
Parallel Printer Int. card	149.00
H. S. Serial Int. card	159.00
Centronics Printer Int. card	179.00
Communication Int. card	179.00
Interger BASIC Firmware Card	155.00

MICROMATION S-100 BOARDS

- Z-64 Board (list $1295) **$1099**
  4Mhz Z80 CPU with 64K
- VIO-F8 (list $495) **439**
  80 × 24 Memory Mapped Video with Serial & Parallel I/O
- DS-F8 (list $495) **429**
  Double density controller
- MD Controller (list $895) **769**
  For shugart SA4004-8
- Multi/User I/O (list $295) **249**
  4 Serial, 2 parallel, Clock

FUTRA COMPANY
P. O. BOX 4380
DEPARTMENT I
TORRANCE, CA 90510
Phone (213) 370-2933

MICROSOFT Z-80 Soft Card

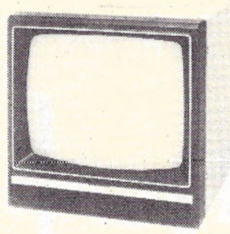


FUTRA PRICE
$259
list $349

Now! Z80-CP/M Capability in your Apple II. Switch between Z80/8080 software and Apple II 6502 software CP/M & Microsoft Basic Incl.

12" SANYO GREEN SCREEN MONITOR

FUTRA PRICE
$274
list $325



800 line resolution monitor with Green Phosphoric Screen for pleasant viewing. Perfect for new 80 by 24 Video Boards
LEEDEX 12" B&W $153

APPLE Accessories

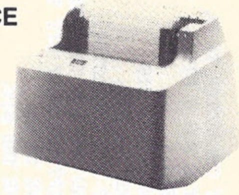
- Novation CAT Modem **$179**
- Apple Clock Card **249**
- ABT Key Pad **115**
- ALF Music Card **239**
- DC Hayes Micromodem II **345**
- Speech Lab Disk Software **189**
- VisiCalc **120**
- CCA Data Management **85**
- Desk Top Plan **85**

DATA SUPPLIES

- 5" ¼" Memorex Diskettes **$27**
  0, 10, or 16 Sector – Box of 10
- 8" Memorex diskettes **$32**
  0 or 32 Sector – Box of 10
- 3 ring Diskette Sleves (10) ... **$6.95**
  Each holds (2) 5" ¼" diskettes

GRAPHICS PAPER TIGER

FUTRA PRICE
$879
list $1095



A high performance dot matrix 96 character printer with Factory Graphics Opt. Serial and Parallel Interface
EMAKO 22 printer $749

Now! 80 by 24 Screen display for your Apple

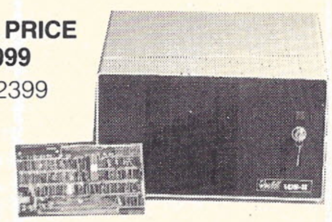
Videx 80 × 24 "Videoterm" with Graphics Rom

FUTRA Price $269
list $369

7 by 9 dot Matrix, upper and lower case with true decenders. 1K firmware incorporates pascal and Basic protocalls. Compatible with Apple peripherials, User definable character set, Inverse display. Super for Language System and Z-80 Softcard.

TARBELL VDS-IID

FUTRA PRICE
$2099
list $2399



A complete double density S-100 Disk Subsystem, Dual 8" drives and controller, attractive cabinet, power supply, CP/M, Tarbell BASIC
VDS-IIDD Double Sided $2449

MOUNTAIN HARWARE for Apple II

- Supertalker **$259**
- Romplus with filter **$175**
- Romwriter **$149**
- INTROL X/10 System **$259**

TERMS OF SALE: Master Charge or Visa add 3% service charge to purchase price. No COD's. Allow 3 weeks for personal checks to clear. Orders under $50.00, add $2.00 for shipping and handling costs. All other orders (unless specified in ad) that are within 50 lbs. limit of U.P.S. will be shipped, no charge. Shipments over 50 lbs. in weight or out of Continental U.S. shipped freight collect. All prices are subject to change and all offers subject to withdrawal without notice.
California residents add 6% or 6.5% if serviced by BART.

A Modem Interface

Continued from Page 98

LISTING 1

MODEM INTERFACE ASSEMBLY LISTING

```
6200           0100 ************************************************************
6200           0110 *   Z80/8080 MODEM INTERFACE   VERSION 1.3                 *
6200           0120 *     N. M. HARRINGTON   NOVEMBER, 1979                    *
6200           0130 ************************************************************
6200           0140 *
6200           0150 *   INITIALIZE SCREEN BUFFER
6200           0160 *-----------------------------------------------------------
6200 CD 55 63  0170        CALL CLEARS         :CLEAR SCREEN.
6203 21 72 63  0180        LD   HL,BUFFER      :INITIALIZE TOP OF DISPLAY
6206 22 72 67  0190        LD   (TOP),HL       :   POINTER.
6209 3E A0     0200        LD   A,BLANK        :FILL BUFFER WITH BLANKS.
620B 0E 10     0210        LD   C,NLINES
620D 06 40     0220 CLRBUF LD   B,NCHARS
620F CD 4E 63  0230        CALL FILMEM
6212 0D        0240        DEC  C
6213 C2 0D 62  0250        JP   NZ,CLRBUF
6216 0E 0F     0260        LD   C,NLINES-1     :MOVE TO BOTTOM LINE ON SCREEN.
6218 06 40     0270 CLOOP  LD   B,NCHARS
621A CD 6A 63  0280 BLOOP  CALL SPACE
621D 05        0290        DEC  B
621E C2 1A 62  0300        JP   NZ,BLOOP
6221 0D        0310        DEC  C
6222 C2 18 62  0320        JP   NZ,CLOOP
6225 21 32 67  0330        LD   HL,LASLIN      :SET INPUT LINE POINTER TO LAST
6228 22 74 67  0340        LD   (INLINE),HL    :   LINE ON SCREEN.
622B CD 45 63  0350        CALL DSCRSR         :DISPLAY CURSOR.
622E           0360 *===========================================================
622E           0370 *   *****  MAIN ROUTINE  *****
622E           0380 *   ALTERNATELY POLL MODEM, THEN KEYBOARD.
622E           0390 *===========================================================
622E DB 02     0400 CYCLE  IN   INMODM         :CHECK MODEM.
6230 E6 01     0410        AND  1              :ANYTHING THERE?
6232 C2 3B 62  0420        JP   NZ,KEYIN       :NO.  GO CHECK KEYBOARD.
6235 CD 88 62  0430        CALL RMODEM         :GET CHARACTER FROM MODEM.
6238 CD A2 62  0440        CALL DCHAR          :DISPLAY IT.
623B DB 00     0450 KEYIN  IN   INKEYB         :NOW CHECK KEYBOARD.
623D FE 00     0460        CP   0              :ANYTHING THERE?
623F CA 2E 62  0470        JP   Z,CYCLE        :NO.  GO AROUND AGAIN.
6242 FE 20     0480        CP   CCHAR          :IS IT A CONTROL CHARACTER?
6244 D2 51 62  0490        JP   NC,STROBE      :NO.  GO CHECK STROBE BIT.
6247 FE 1B     0500        CP   ESCAPE         :IS IT "ESCAPE" KEY?
6249 CA 00 05  0510        JP   Z,MONITR       :YES.  EXIT TO OP SYSTEM.
624C FE 0D     0520        CP   CR             :IS IT "RETURN" KEY?
624E C2 5B 62  0530        JP   NZ,WRITEM      :NO.  SEND CHARACTER TO MODEM.
6251 FE 80     0540 STROBE CP   MSBIT          :HAS KEY JUST BEEN PRESSED?
6253 DA 2E 62  0550        JP   C,CYCLE        :NO.  IGNORE IT.
```

```
62BD 57        1150 WMODEM LD   D,A            :PUT CHARACTER IN D.
62BE 1E 08     1160        LD   E,8D           :SET E FOR 8 BITS.
62C0 AF        1170        XOR  A              :CLEAR A.
62C1 D3 02     1180        OUT  OTMODM         :SEND START BIT.
62C3 CD DD 62  1190        CALL DELAY          :TIME THE BIT.
62C6 7A        1200 OUTIT  LD   A,D            :GET THE CHARACTER.
62C7 D3 02     1210        OUT  OTMODM         :SEND THE LSB TO MODEM.
62C9 0F        1220        RRCA                :SHIFT THE NEXT BIT TO LSB.
62CA 57        1230        LD   D,A            :SAVE IN D.
62CB CD DD 62  1240        CALL DELAY          :TIME THE BIT.
62CE 1D        1250        DEC  E              :DONE WITH 8 BITS?
62CF C2 C6 62  1260        JP   NZ,OUTIT       :NO.  GO SEND NEXT BIT.
62D2 3E 01     1270        LD   A,1            :DONE.  SEND 2 STOP BITS.
62D4 D3 02     1280        OUT  OTMODM
62D6 CD DD 62  1290        CALL DELAY
62D9 CD DD 62  1300        CALL DELAY
62DC C9        1310        RET
62DD           1320 *-----------------------------------------------------------
62DD           1330 *   BAUD RATE TIMING LOOP
62DD           1340 *-----------------------------------------------------------
62DD 0E 04     1350 DELAY  LD   C,TIM1
62DF 06 92     1360 DELAY1 LD   B,TIM2
62E1 05        1370 STALL  DEC  B
62E2 C2 E1 62  1380        JP   NZ,STALL
62E5 0D        1390        DEC  C
62E6 C2 DF 62  1400        JP   NZ,DELAY1
62E9 C9        1410        RET
62EA           1420 *-----------------------------------------------------------
62EA           1430 *   SUBROUTINE TO PROCESS A LINEFEED
62EA           1440 *-----------------------------------------------------------
62EA 2A 72 67  1450 LNFEED LD   HL,(TOP)       :MOVE TOP OF DISPLAY POINTER
62ED CD 25 63  1460        CALL MOVELP         :   DOWN A LINE.
62F0 22 72 67  1470        LD   (TOP),HL
62F3 2A 74 67  1480        LD   HL,(INLINE)    :MOVE INPUT LINE POINTER DOWN
62F6 CD 25 63  1490        CALL MOVELP         :   A LINE.
62F9 22 74 67  1500        LD   (INLINE),HL
62FC 3E A0     1510        LD   A,BLANK        :BLANK OUT BOTTOM DISPLAY LINE
62FE 06 40     1520        LD   B,NCHARS       :   IN BUFFER.
6300 CD 4E 63  1530        CALL FILMEM
6303 CD 0A 63  1540        CALL DSPLAY         :SCROLL THE DISPLAY.
6306 CD 45 63  1550        CALL DSCRSR         :DISPLAY CURSOR AT BOTTOM LINE.
6309 C9        1560        RET
630A           1570 *-----------------------------------------------------------
630A           1580 *   SUBROUTINE TO PUT ENTIRE DISPLAY BUFFER ON SCREEN
630A           1590 *-----------------------------------------------------------
630A CD 55 63  1600 DSPLAY CALL CLEARS         :FIRST CLEAR SCREEN.
630D 2A 72 67  1610        LD   HL,(TOP)       :POINT TO TOP OF DISPLAY.
6310 0E 0F     1620        LD   C,NLINES-1     :SET C AND B FOR NLINES -1 OF
6312 06 40     1630 LINE   LD   B,NCHARS       :   NCHARS EACH.
6314 7E        1640 CHAR   LD   A,M            :GET CHARACTER FROM BUFFER.
6315 CD 6C 63  1650        CALL TVOUT          :DISPLAY IT.
6318 23        1660        INC  HL             :ADVANCE POINTER TO NEXT CHAR.
6319 05        1670        DEC  B              :DO UNTIL END OF LINE.
631A C2 14 63  1680        JP   NZ,CHAR
631D CD 29 63  1690        CALL TESTLP         :POINT TO NEXT LINE IN BUFFER.
6320 0D        1700        DEC  C              :DONE WITH NLINES -1?
6321 C2 12 63  1710        JP   NZ,LINE        :NO.  GO DO ANOTHER LINE.
6324 C9        1720        RET
```

```
6256 FE FF      0560          CP   DEL            :IS IT "DELETE" KEY?
6258 CA 61 62   0570          JP   Z,RUBOUT       :YES.  GO ERASE CHARACTER.
625B CD BD 62   0580 WRITEM CALL WMODEM           :SEND CHARACTER TO MODEM.
625E C3 2E 62   0590          JP   CYCLE          :GO AROUND AGAIN.
6261            0600 *-----------------------------------------------------------
6261            0610 *   ROUTINE TO ERASE A BAD INPUT CHARACTER
6261            0620 *-----------------------------------------------------------
6261 CD 39 63   0630 RUBOUT CALL ERASE            :FIRST ERASE CURSOR FROM SCREEN.
6264 CD 39 63   0640          CALL ERASE          :THEN ERASE PRECEDING CHARACTER.
6267 2A 76 67   0650          LD   HL,(CURSOR)    :BACK UP THE CURSOR.
626A 2B         0660          DEC  HL
626B 36 A0      0670          LD   M,BLANK        :BLANK OUT CHARACTER IN BUFFER.
626D CD 45 63   0680          CALL DSCRSR
6270 3E FF      0690          LD   A,DEL          :SEND "DELETE" CHARACTER TO
6272 CD BD 62   0700          CALL WMODEM         :   MODEM.
6275 2E 03      0710          LD   L,3            :NOW INTERCEPT NEXT 3 CHARACTERS:
6277 DB 02      0720 INCEPT IN   INMODM           :   BACKSLASH, DELETED CHARACTER,
6279 E6 01      0730          AND  1              :   BACKSLASH.
627B C2 77 62   0740          JP   NZ,INCEPT
627E CD 88 62   0750          CALL RMODEM
6281 2D         0760          DEC  L
6282 C2 77 62   0770          JP   NZ,INCEPT
6285 C3 3B 62   0780          JP   KEYIN          :GO GET CORRECT KEYIN.
6288            0790 *-----------------------------------------------------------
6288            0800 *   SUBROUTINE TO READ A CHARACTER FROM MODEM
6288            0810 *-----------------------------------------------------------
6288 0E 06      0820 RMODEM LD   C,TIM0           :WAIT FOR CENTER OF FIRST DATA
628A CD DF 62   0830          CALL DELAY1         :   BIT TO ARRIVE.
628D 11 08 00   0840          LD   DE,8D          :CLEAR D.  SET E FOR 8 BITS.
6290 DB 02      0850 NEXT   IN   INMODM           :GET INPUT FROM MODEM.
6292 E6 01      0860          AND  1              :MASK LSB.
6295 82         0870          ADD  D              :ADD IN PREVIOUS BITS READ.
6296 0F         0880          RRCA                :SHIFT FOR NEXT INPUT.
6297 57         0890          LD   D,A            :SAVE PARTIAL BYTE IN D.
6298 CD DD 62   0900          CALL DELAY          :WAIT FOR CENTER OF NEXT BIT.
629B 1D         0910          DEC  E              :DONE WITH 8 BITS?
629C C2 90 62   0920          JP   NZ,NEXT        :NO.  GO GET NEXT BIT.
629F F6 80      0930          OR   80H            :DONE.  SET MSB.
62A1 C9         0940          RET
62A2            0950 *-----------------------------------------------------------
62A2            0960 *   SUBROUTINE TO DISPLAY CHARACTER ON SCREEN
62A2            0970 *-----------------------------------------------------------
62A2 FE 8D      0980 DCHAR  CP   LF               :IS IT "LINEFEED"?
62A4 CC EA 62   0990          CALL Z,LNFEED       :YES.  GO DO IT.
62A7 FE A0      1000          CP   BLANK          :IS IT A CONTROL CHARACTER?
62A9 D8         1010          RET  C              :YES.  IGNORE IT.
62AA FE FF      1020          CP   DEL            :IS IT "DELETE"?
62AC C8         1030          RET  Z              :YES.  IGNORE IT.
62AD 2A 76 67   1040          LD   HL,(CURSOR)    :STORE CHARACTER IN SCREEN BUFFER
62B0 77         1050          LD   M,A            :   AND DISPLAY IT ON SCREEN IN
62B1 CD 3F 63   1060          CALL BSPACE         :   PLACE OF CURSOR.
62B4 7E         1070          LD   A,M
62B5 CD 6C 63   1080          CALL TVOUT
62B8 23         1090          INC  HL             :ADVANCE CURSOR AND DISPLAY IT.
62B9 CD 45 63   1100          CALL DSCRSR
62BC C9         1110          RET
62BD            1120 *-----------------------------------------------------------
62BD            1130 *   SUBROUTINE TO WRITE A CHARACTER TO MODEM
62BD            1140 *-----------------------------------------------------------
```

```
6325            1730 *-----------------------------------------------------------
6325            1740 *   SUBROUTINE TO ADVANCE BUFFER POINTER TO NEXT LINE
6325            1750 *-----------------------------------------------------------
6325 11 40 00   1760 MOVELP LD   DE,NCHARS        :ADD NCHARS TO CURRENT POINTER
6328 19         1770          ADD  HL,DE          :   LOCATION.
6329 11 71 67   1780 TESTLP LD   DE,ENDBUF        :NOW CHECK IF THIS PUTS POINTER
632C 7C         1790          LD   A,H            :   PAST THE END OF THE ACTUAL
632D BA         1800          CP   D              :   BUFFER REGION.
632E D8         1810          RET  C
632F C2 35 63   1820          JP   NZ,WRAP
6332 7D         1830          LD   A,L
6333 BB         1840          CP   E
6334 D8         1850          RET  C              :OK.  RETURN.
6335 21 72 63   1860 WRAP   LD   HL,BUFFER        :WRAP AROUND TO START OF ACTUAL
6338 C9         1870          RET                 :   BUFFER REGION.
6339            1880 *-----------------------------------------------------------
6339            1890 *   SUBROUTINE TO ERASE A CHARACTER
6339            1900 *-----------------------------------------------------------
6339 CD 3F 63   1910 ERASE  CALL BSPACE           :BACK DISPLAY UP OVER LAST CHAR.
633C CD 6A 63   1920          CALL SPACE          :BLANK IT OUT.
633F            1930 *                            :DROP THRU TO BACK OVER BLANK.
633F            1940 *-----------------------------------------------------------
633F            1950 *   SUBROUTINE TO BACKSPACE SCREEN DISPLAY
633F            1960 *-----------------------------------------------------------
633F 3E 06      1970 BSPACE LD   A,BAKSPC         :LOAD BACKSPACE CHARACTER.
6341 CD 6C 63   1980          CALL TVOUT          :SEND IT TO SCREEN.
6344 C9         1990          RET
6345            2000 *-----------------------------------------------------------
6345            2010 *   SUBROUTINE TO DISPLAY CURSOR
6345            2020 *-----------------------------------------------------------
6345 22 76 67   2030 DSCRSR LD   (CURSOR),HL      :SAVE PRESENT CURSOR LOCATION.
6348 3E DF      2040          LD   A,USCORE       :LOAD UNDERSCORE CHARACTER.
634A CD 6C 63   2050          CALL TVOUT          :DISPLAY IT.
634D C9         2060          RET
634E            2070 *-----------------------------------------------------------
634E            2080 *   SUBROUTINE TO FILL A BLOCK OF MEMORY
634E            2090 *-----------------------------------------------------------
634E 77         2100 FILMEM LD   M,A              :PUT CHARACTER IN MEMORY.
634F 23         2110          INC  HL             :ADVANCE MEMORY POINTER.
6350 05         2120          DEC  B              :DO IT B TIMES.
6351 C2 4E 63   2130          JP   NZ,FILMEM
6354 C9         2140          RET
6355            2150 *-----------------------------------------------------------
6355            2160 *   SUBROUTINE TO CLEAR SCREEN AND HOME TO UPPER LEFT
6355            2170 *-----------------------------------------------------------
6355 3E 7F      2180 CLEARS LD   A,RUB            :LOAD DELETE CHARACTER.
6357 CD 6C 63   2190          CALL TVOUT          :HOME TO UPPER LEFT CORNER.
635A 06 10      2200          LD   B,NLINES       :OUTPUT NULL CHARACTERS TO
635C 0E 40      2210 CLOAD  LD   C,NCHARS         :   ERASE SCREEN.
635E CD 6A 63   2220 BLANKM CALL SPACE
6361 0D         2230          DEC  C
6362 C2 5E 63   2240          JP   NZ,BLANKM
6365 05         2250          DEC  B
6366 C2 5C 63   2260          JP   NZ,CLOAD
6369 C9         2270          RET
636A            2280 *-----------------------------------------------------------
636A            2290 *   SUBROUTINE TO DISPLAY A BLANK OR A CHARACTER
636A            2300 *-----------------------------------------------------------
636A 3E A0      2310 SPACE  LD   A,BLANK          :LOAD A BLANK.
```

```
636C D3 00     TVOUT  OUT  OTSCRN          :DISPLAY CHARACTER IN A.
636E AF               XOR  A               :CLEAR A.
636F D3 00            OUT  OTSCRN          :OUTPUT A NULL CHARACTER.
6371 C9               RET
6372           *===================================================
6372           *   SYMBOL DEFINITIONS
6372           *---------------------------------------------------
6372           BLANK  EQU  0A0H            :BLANK CHARACTER
6372           CR     EQU  0DH             :CARRIAGE RETURN
6372           LF     EQU  8DH             :LINE FEED (CR W MSB ON)
6372           RUB    EQU  7FH             :RUBOUT CHARACTER
6372           DEL    EQU  0FFH            :DELETE CHARACTER
6372           USCORE EQU  0DFH            :UNDERSCORE CHARACTER
6372           ESCAPE EQU  1BH             :ESCAPE CHARACTER
6372           BAKSPC EQU  6               :BACKSPACE CONTROL CHARACTER
6372           MSBIT  EQU  80H             :MOST SIGNIFICANT BIT
6372           MONITR EQU  0500H           :ADDRESS OF MONITOR
6372           CCHAR  EQU  20H             :1ST CHAR BEYOND CONTROL CHARS.
6372           *---------------------------------------------------
6372           *   I/O PORT DEFINITIONS
6372           *---------------------------------------------------
6372           INKEYB EQU  0               :KEYBOARD INPUT PORT
6372           OTSCRN EQU  0               :VIDEO OUTPUT PORT
6372           INMODM EQU  2               :MODEM INPUT PORT
6372           OTMODM EQU  2               :MODEM OUTPUT PORT
6372           *---------------------------------------------------
6372           *   TIMING CONSTANTS
6372           *---------------------------------------------------
6372           TIM1   EQU  4               :TIMING LOOP CONSTANTS FOR 300
6372           TIM2   EQU  146D            :   BAUD WITH 2.5 MHZ CLOCK
6372           TIM0   EQU  6               :TIM1*3/2
6372           *---------------------------------------------------
6372           *   VIDEO DISPLAY BUFFER STORAGE
6372           *---------------------------------------------------
6372           NLINES EQU  16D             :NUMBER OF LINES ON SCREEN
6372           NCHARS EQU  64D             :NUMBER OF CHARACTERS PER LINE
6372           NBUF   EQU  1024D           :NLINES*NCHARS
6372           BUFFER DS   NBUF            :STORAGE FOR NBUF CHARACTERS
6772           ENDBUF EQU  BUFFER+NBUF-1   :ADDRESS OF LAST LOC IN BUFFER
6772           LASLIN EQU  BUFFER+NBUF-NCHARS
6772           TOP    DS   2               :POINTER TO 1ST LINE OF DISPLAY
6774           INLINE DS   2               :POINTER TO INPUT (BOTTOM) LINE
6776           CURSOR DS   2               :POINTER TO CURSOR LOCATION
```

```
SYMB
CLRBUF 620D   CLOOP  6218   BLOOP  621A   CYCLE  622E
KEYIN  623B   STROBE 6251   WRITEM 625B   RUBOUT 6261
INCEPT 6277   RMODEM 6288   NEXT   6290   DCHAR  62A2
WMODEM 62BD   OUTIT  62C6   DELAY  62DD   DELAY1 62DF
STALL  62E1   LNFEED 62EA   DSPLAY 630A   LINE   6312
CHAR   6314   MOVELP 6325   TESTLP 6329   WRAP   6335
ERASE  6339   BSPACE 633F   DSCRSR 6345   FILMEM 634E
CLEARS 6355   CLOAD  635C   BLANKM 635E   SPACE  636A
TVOUT  636C   BLANK  00A0   CR     000D   LF     008D
RUB    007F   DEL    00FF   USCORE 00DF   ESCAPE 001B
BAKSPC 0006   MSBIT  0080   MONITR 0500   CCHAR  0020
INKEYB 0000   OTSCRN 0000   INMODM 0002   OTMODM 0002
TIM1   0004   TIM2   0092   TIM0   0006   NLINES 0010
NCHARS 0040   NBUF   0400   BUFFER 6372   ENDBUF 6771
LASLIN 6732   TOP    6772   INLINE 6774   CURSOR 6776
```

LISTING 2

BASIC PROGRAM TO COMPUTE TIMING LOOP CONSTANTS

```
10 INPUT "ENTER CLOCK FREQUENCY IN MEGAHERTZ: ", F
20 INPUT "ENTER BAUD RATE: ", R
30 C = 0
40 PRINT
50 PRINT "  C        B         %ERROR"
60 PRINT "------ ------- ----------"
70 FOR I=1 TO 10
80 C = C + 2
90 IF C>255 THEN 170
100 B = INT(((1.E6*F/R - 73)/C - 21)/14 + .5)
110 IF B>255 THEN 80
120 E = 100*((C*(14*B + 21) + 73)*1.E-6*R/F - 1)
130 PRINT C, B, E
140 NEXT I
150 INPUT "DO YOU WANT MORE (Y OR N)", A$
160 IF A$="Y" THEN 40
170 END
```

SAMPLE RUN

SAMPLE RUN OF BASIC TIMING CONSTANT PROGRAM

```
READY
RUN

ENTER CLOCK FREQUENCY IN MEGAHERTZ: 2.5
ENTER BAUD RATE: 300

  C       B        %ERROR
------ ------- ----------
 4       146      -.004
 6        97     .164
 8        72     -.34
 10        58     .836
 12        48     .668
 14        41     .836
 16        35     -1.012
 18        31     -.844
 20        28     -.004
 22        25     -1.18
DO YOU WANT MORE (Y OR N)N
READY
```

An Interrupt Driven

Continued from Page 107

PROGRAM LISTING

```
; BIOS:    USER I/O FOR LIFEBOAT CP/M
;          ON NORTH STAR DISK
; INTERRUPT-DRIVEN KEYBOARD BUFFER
; COMPUTIME INITIALIZATION

; PROGRAMMED FOR AN 8080 MICROPROCESSOR
; BY ALAN R. MILLER
; NEW MEXICO TECH, SOCORRO, NM  87801
;  505-835-5619            APR 1, 1979

; TERMINAL DEVICES SUPPORTED:

;               ADDRESS (HEX)      COMMENT
;  CONSOLE          10/11          0 NULLS
;  LIST DEVICE      12/13          TELETYPE
;  MODEM            14/15          HALF OR FULL DUPLEX

; IMPLEMENTATION OF IOBYTE TO SELECT PERIPHERALS

;  CONSOLE:     INPUT   OUTPUT
;       0       CRT     CRT       = CRT
;       1       CRT     LIST      = LST
;       2       LIST    LIST      = LPT
;       3       LIST    CRT       = LCR

;  READER:  (USES CONSOLE INPUT)
;       0       CRT               = LST
;       4       CRT               = ACR
;       8       LIST              = MOD
;       0CH     LIST              = CRT

;  PUNCH:
;       0       MODEM, FULL DUPLEX       = MOD
;       10H     MODEM, HALF DUPLEX       = MO2
;       20H     PUNCH                    = PCH
;       30H     CONSOLE                  = CRT

;  LIST:
;       0       LIST                     = LST
;       40H     CONSOLE                  = CRT
;       80H     MODEM, FULL DUPLEX       = MOD
;       C0H     MODEM, HALF DUPLEX       = MO2

; FOR MODEM, USE PIP
;   A>PIP   PUN:=B:FILENAME
```

```
0000 =         FALSE   EQU     0
FFFF =         TRUE    EQU     NOT FALSE

FFFF =         SHORT   EQU     TRUE      ;HOMEMADE BOARD
FFFF =         PERCOM  EQU     TRUE      ;PERCOM VS. MITS MODEM

0034 =         MSIZE   EQU     52        ;MEMORY SIZE, DECIMAL K

D400 =         ORIGIN  EQU     MSIZE * 1024 + 400H
D400           ORG     ORIGIN
               ; ****************************************
FFFF =         INTRM   EQU     TRUE      ;PATCH INTERRUPTS

                       IF      INTRM     ;INTERRUPTS
0095 =         STOPC   EQU     95H       ;CONSOLE STOP BITS
D1E5 =         INTRP   EQU     ORIGIN-21BH ;INTERRUPT PATCH

               ; CONSOLE BUFFER LOCATION

EE00 =         CPNTR   EQU     0EE00H    ;COMPUTER POINTER
EE02 =         CCNT    EQU     CPNTR+2   ;BUFFER COUNT
EE03 =         KCNT    EQU     CCNT+1    ;KEYBRD BUFF COUNT
EE04 =         KPNTR   EQU     KCNT+1    ;KEYBOARD POINTER
EE06 =         BUFF    EQU     KPNTR+2   ;INPUT BUFFER
00FE =         VI      EQU     0FEH      ;VI BOARD ADDR
0005 =         LEV     EQU     5         ;INTERR LEVEL
00F0 =         RESET   EQU     0F0H      ;RESET BYTE
               ; ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                       ELSE
               STOPC   EQU     15H       ;CONSOLE STOP BITS
                       ENDIF
0000 =         NNULS   EQU     0         ;NUMBER OF CONSOLE NULLS
FFFF =         CLOCK   EQU     TRUE      ;COMPUTIME VERSION
0010 =         CSTAT   EQU     10H       ;CONSOLE STATUS
0011 =         CDATA   EQU     CSTAT+1   ;CONSOLE DATA
0012 =         LSTAT   EQU     12H       ;LIST STATUS
0013 =         LDATA   EQU     LSTAT+1   ;LIST DATA
0006 =         TSTAT   EQU     6         ;MAGNETIC TAPE
0007 =         TDATA   EQU     TSTAT+1
0014 =         MSTAT   EQU     14H       ;MODEM
0015 =         MDATA   EQU     MSTAT+1
00C4 =         ADATA   EQU     0C4H      ;CLOCK A DATA
00C5 =         ACONT   EQU     ADATA+1   ;CLOCK A CONTROL
00C6 =         BDATA   EQU     ADATA+2   ;CLOCK B DATA
00C7 =         BCONT   EQU     ADATA+3   ;CLOCK B CONTROL
                       IF      PERCOM
0040 =         MIMSK   EQU     40H       ;MODEM INPUT MASK
                       ELSE
               MIMSK   EQU     1
                       ENDIF
0001 =         CIMSK   EQU     1         ;CONSOLE INPUT MASK
0001 =         LIMSK   EQU     1         ;LIST INPUT MASK
0001 =         TIMSK   EQU     1         ;MAG TAPE INPUT MASK
0080 =         MOMSK   EQU     MIMSK*2   ;MODEM OUTPUT MASK
```

```
0002 =          COMSK   EQU     CIMSK*2 ;CONSOLE OUTPUT MASK
0002 =          LOMSK   EQU     LIMSK*2 ;LIST OUTPUT MASK
0080 =          TOMSK   EQU     80H     ;TAPE OUTPUT MASK
00FF =          SSW     EQU     255     ;FRONT-PANEL SWITCHES
0011 =          STOPL   EQU     11H     ;LIST = 2  STOPS BITS
000D =          CR      EQU     13      ;CARRIAGE RET
000A =          LF      EQU     10      ;LINE FEED
000C =          FFEED   EQU     12      ;FORM FEED IS ^L
0003 =          CTRC    EQU     3       ;^C
0004 =          CTRD    EQU     4       ;^D

0005 =          CTRE    EQU     5       ;^E
000F =          CTRO    EQU     15      ;^O
0011 =          CTRQ    EQU     17      ;^Q
0013 =          CTRS    EQU     19      ;^S
0003 =          IOBYTE  EQU     3       ;I/O CONFIGURATION

                OUTPUT  MACRO   Y,Z     ;;MACRO FOR OUTPUT
                        LOCAL   LOOP    ;;INTERNAL LABEL
                LOOP:   IN      Y&STAT  ;CHECK STATUAS
                        ANI     Y&OMSK  ;MASK FOR OUTPUT
                        J&Z     LOOP    ;LOOP UNTIL READY
                        MOV     A,C     ;GET BYTE
                        OUT     Y&DATA  ;OUTPUT IT
                        ENDM

D400 C345D4     START:  JMP     INIT    ;INITIALIZATION
D403 C3F8D4             JMP     CONST   ;CONSOLE STATUS
D406 C312D4             JMP     CONIN   ;CONSOLE INPUT
D409 C328D5             JMP     CONOUT  ;CONSOLE OUTPUT
D40C C358D5             JMP     LOUT    ;LIST OUTPUT
D40F C33CD5             JMP     PUNCH   ;PUNCH OUTPUT

                ; USE CONIN FOR READER

                ; CONSOLE-INPUT ROUTINE

D412 3A0300     CONIN:  LDA     IOBYTE  ;GET I/O BYTE
D415 E602               ANI     2       ;CHECK BIT 1
D417 C2A5D5             JNZ     LIN     ;LIST INPUT

                ; *******************************************
                        IF      INTRM   ;INTERRUPTS

                ; GET INPUT FROM KEYBOARD BUFFER
                ; INSTEAD OF FROM CONSOLE

D41A E5                 PUSH    H
D41B 2A02EE     CIN3:   LHLD    CCNT    ;BOTH COUNTS
D41E 7C                 MOV     A,H
D41F 95                 SUB     L       ;SAME?
D420 CA1BD4             JZ      CIN3    ;KEEP TRYING
D423 F3                 DI              ;HOLD OFF
D424 2102EE             LXI     H,CCNT  ;COMPUTER COUNT
D427 34                 INR     M       ;INCREMENT IT
D428 2A00EE             LHLD    CPNTR   ;COMPUTER PNTR
D42B 7E                 MOV     A,M     ;GET BYTE

                ; RESET BOTH POINTERS IF CARR RET FOUND
```

```
D45E 3E77               MVI     A,77H
D460 D3C6               OUT     BDATA
D462 3E14               MVI     A,14H   ;NO CLK INTRR
D464 D3C5               OUT     ACONT
D466 3E04               MVI     A,4

D468 D3C7               OUT     BCONT
                        ENDIF           ;CLOCK

                ; ***************************************
                        IF      INTRM

                ; PATCH RST LOCATION TO JUMP TO KEYBD
                ; AND LIFEBOAT SYSTEM WITH EI

D46A F3                 DI
D46B 3EFB               MVI     A,0FBH  ;EI INSTRUCTION
D46D 32E5D1             STA     INTRP   ;PATCH LIFEBOAT
D470 3EC3               MVI     A,0C3H  ;JMP INSTR
D472 322800             STA     8*LEV   ;PATCH RST
D475 E5                 PUSH    H
D476 21AAD4             LXI     H,KEYBD ;INTERR ENTRY
D479 222900             SHLD    8*LEV+1 ;JUMP HERE
D47C 2106EE             LXI     H,BUFF  ;BUFFER ADDR
D47F 2204EE             SHLD    KPNTR   ;RESET POINTERS
D482 2200EE             SHLD    CPNTR
D485 210000             LXI     H,0     ;2 ZEROS
D488 2202EE             SHLD    CCNT    ;ZERO THE COUNTS
D48B E1                 POP     H
                        IF      NOT SHORT   ;INTERR BOARD
                        MVI     A,RESET
                        OUT     VI      ;RESET VI BOARD
                        ENDIF
D48C FB                 EI
                ; ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                        ELSE            ;NO INTERRUPTS
                        DI
                        ENDIF

D48D 3AE5D5             LDA     IFLAG   ;INITILIZATION FLAG
D490 FE23               CPI     '#'     ;# MEANS INITIALIZED
D492 C8                 RZ              ;SO SKIP THE REST

                ; COLD-START INITIALIZATION

D493 3E23               MVI     A,'#'
D495 32E5D5             STA     IFLAG   ;SET FLAG
D498 AF                 XRA     A       ;GET A ZERO
D499 320300             STA     IOBYTE  ;SET TO STANDARD I/O
D49C 21D2D5             LXI     H,SIGNON  ;SIGNON MESSAGE
D49F 7E         MESG:   MOV     A,M     ;GET MESSAGE
D4A0 B7                 ORA     A       ;ZERO FOR LAST BYTE
D4A1 C8                 RZ              ;DONE
D4A2 4F                 MOV     C,A
D4A3 CD31D5             CALL    CONW    ;PRINT SIGN-ON MESSAGE
D4A6 23                 INX     H
D4A7 C39FD4             JMP     MESG    ;NEXT CHARACTER
```

```
D42C 23                 INX     H       ;BUMP POINTER
D42D 2200EE             SHLD    CPNTR   ;SAVE IT
D430 FE0D               CPI     CR      ;CARRIAGE RET?
D432 C242D4             JNZ     CIN4    ;NO
D435 2A02EE             LHLD    CCNT    ;GET BOTH COUNTS
D438 7C                 MOV     A,H
D439 95                 SUB     L       ;DIFFERENCE
D43A C240D4             JNZ     CIN5    ;NOT SAME

                ; RESET BOTH POINTERS TO ZERO

D43D CDE8D4             CALL    RSETP
D440 3E0D       CIN5:   MVI     A,CR    ;RESTORE CR
D442 E1         CIN4:   POP     H
D443 FB                 EI              ;READY FOR MORE
D444 C9                 RET
                ; ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                        ELSE            ;NOT INTERRUPTS
                CONSIN: IN      CSTAT   ;CONSOLE INPUT
                        RAR
                        JNC     CONSIN
                        IN      CDATA
                        ANI     7FH     ;MASK PARITY
                        RET
                        ENDIF

                ; INITIALIZE INTERFACE CHIPS EACH PASS
                ; AND IOBYTE ON COLD START

D445 3E03       INIT:   MVI     A,3
D447 D310               OUT     CSTAT   ;RESET ACIA
D449 D312               OUT     LSTAT

                        IF      NOT PERCOM
                        OUT     MSTAT   ;RESET MITS MODEM
                        MVI     A,STOPC
                        OUT     MSTAT   ;SET MODEM STOP BITS
                        ENDIF

D44B 3E95               MVI     A,STOPC
D44D D310               OUT     CSTAT   ;CONSOLE STOP BITS
D44F 3E11               MVI     A,STOPL
D451 D312               OUT     LSTAT   ;LIST STOP BITS
D453 AF                 XRA     A       ;GET A ZERO
                ;       STA     IOBYTE  ;SET TO STD I/O ON ^C

                        IF      PERCOM
D454 D314               OUT     MSTAT   ;RESET PECOM MODEM
                        ENDIF

                ; INITIALIZE COMPUTIME BOARD

                        IF      CLOCK
D456 D3C5               OUT     ACONT   ;RESET CLOCK PORT A
D458 D3C7               OUT     BCONT   ;RESET PORT B
D45A 3E70               MVI     A,70H
D45C D3C4               OUT     ADATA
```

```
                ; ******************************************
                        IF      INTRM

                ; INTERRUPT ENTRY FOR KEYBOARD INPUT

D4AA F5         KEYBD:  PUSH    PSW
D4AB DB10               IN      CSTAT   ;CONSOLE STATUS
D4AD 1F                 RAR
D4AE D2DFD4             JNC     KEY2    ;NOT READY
D4B1 DB11               IN      CDATA   ;GET DATA
D4B3 E67F               ANI     7FH     ;MASK PARITY

                ; CHECK FOR ^S, ^Q SCROLL CONTROL

D4B5 FE13               CPI     CTRS    ;^S
D4B7 C2CCD4             JNZ     KEY3    ;NO
D4BA DB10       KEY4:   IN      CSTAT   ;CHECK KEYBOARD
D4BC 1F                 RAR             ;READY?
D4BD D2BAD4             JNC     KEY4    ;LOOP UNTIL READY
D4C0 DB11               IN      CDATA   ;GET BYTE
D4C2 E67F               ANI     7FH     ;STRIP PARITY
D4C4 FE11               CPI     CTRQ    ;^Q?
D4C6 C2BAD4             JNZ     KEY4    ;NO
D4C9 C3DFD4             JMP     KEY2

D4CC E5         KEY3:   PUSH    H
D4CD FE04               CPI     CTRD    ;^D KILL
D4CF CAE2D4             JZ      KEY6
D4D2 2A04EE             LHLD    KPNTR   ;BUFFER POINTER
D4D5 77                 MOV     M,A     ;PUT IT THERE
D4D6 23                 INX     H       ;INR POINTER
D4D7 2204EE             SHLD    KPNTR   ;SAVE POINTER
D4DA 2103EE             LXI     H,KCNT  ;GET COUNT
D4DD 34                 INR     M       ;INCREMENT IT
D4DE E1         KEY5:   POP     H

                KEY2:
                        IF      NOT SHORT  ;INTER BOARD
                        MVI     A,RESET
                        OUT     VI      ;RESET VI BOARD
                        ENDIF

D4DF F1                 POP     PSW
D4E0 FB                 EI
D4E1 C9                 RET

D4E2 CDE8D4     KEY6:   CALL    RSETP   ;RESET POINTERS
D4E5 C3DED4             JMP     KEY5

                ; RESET BOTH POINTERS TO START

D4E8 210000     RSETP:  LXI     H,0
D4EB 2202EE             SHLD    CCNT    ;ZERO BOTH
D4EE 2106EE             LXI     H,BUFF
D4F1 2204EE             SHLD    KPNTR   ;RESET PNTRS
D4F4 2200EE             SHLD    CPNTR
D4F7 C9                 RET
                ; ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
                    ENDIF               ;INTERR

            ; ROUTINE TO CHECK CONSOLE INPUT STATUS

D4F8 3A0300 CONST:  LDA     IOBYTE  ;CHECK I/O BYTE
D4FB E602           ANI     2       ;CHECK BIT 1
D4FD C21DD5         JNZ     LISST   ;LIST INPUT

            ; ****************************************
                    IF      INTRM

            ; CHECK INPUT BUFFER RATHER THEN KEYBOARD

D500 E5             PUSH    H
D501 2A02EE         LHLD    CCNT    ;BOTH COUNTS
D504 7C             MOV     A,H
D505 95             SUB     L       ;DIFFERENCE
D506 E1             POP     H
D507 C8             RZ              ;NO INPUT

D508 E5             PUSH    H
D509 2A00EE         LHLD    CPNTR   ;COMPUTER PNTR
D50C 7E             MOV     A,M     ;NEXT CHAR
D50D E1             POP     H
D50E FE03           CPI     CTRC    ;^C? MBASIC
D510 CA1AD5         JZ      QUIT    ;YES
D513 FE05           CPI     CTRE    ;^E? XBASIC
D515 CA1AD5         JZ      QUIT    ;YES
D518 AF             XRA     A       ;GET ZERO
D519 C9             RET
            ; ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                    ELSE    ;NO INTERR
                    IN      CSTAT   ;GET STATUS
                    ANI     1       ;MASK FOR INPUT
                    RZ              ;INPUT NOT READY
                    ENDIF           ;INTERR

D51A 3EFF   QUIT:   MVI     A,TRUE  ;FF HEX MEANS READY
D51C C9             RET

D51D DB12   LISST:  IN      LSTAT   ;LIST INPUT STATUS
D51F E601           ANI     1
D521 C8             RZ              ;INPUT NOT READY
D522 3EFF           MVI     A,TRUE  ;FF HEX MEANS READY
D524 C9             RET

            ; HALF DUPLEX, SEND TO BOTH CONSOLE AND MODEM

D525 CD8ED5 MHALF:  CALL    MOUT    ;OUTPUT TO MODEM

            ;  CONSOLE OUTPUT ROUTINE

D528 3A0300 CONOUT: LDA     IOBYTE  ;CHECK I/O BYTE
D52B E603           ANI     3       ;BITS 0,1 FOR CONSOLE
D52D B7             ORA     A       ;CHECK PARITY
D52E E258D5         JPO     LOUT    ;LIST OUTPUT
D565 DAAED5         JC      MFULL   ;MODEM, FULL DUPLEX
D568 C331D5         JMP     CONW    ;SEND TO CONSOLE

            ; LIST OUTPUT TO LIST DEVICE.
            ; FORM-FEED (^L) SIMULATES PAGE EJECT.

            LIST:   OUTPUT  L,Z
D56B+DB12   ??0003: IN      LSTAT   ;CHECK STATUAS
D56D+E602           ANI     LOMSK   ;MASK FOR OUTPUT
D56F+CA6BD5         JZ      ??0003  ;LOOP UNTIL READY
D572+79             MOV     A,C     ;GET BYTE
D573+D313           OUT     LDATA   ;OUTPUT IT
D575 FE0C           CPI     FFEED   ;FORM FEED?
D577 C0             RNZ             ;NO
D578 CD80D5         CALL    LSKIP   ;SKIP FOUR LINES
D57B 0E2D           MVI     C,'-'   ;MARK PLACE TO
D57D CD6BD5         CALL    LIST    ; FOLD PAPER
D580 C5     LSKIP:  PUSH    B
D581 0E0A           MVI     C,LF
D583 0604           MVI     B,4
D585 CD6BD5 LSKP2:  CALL    LIST
D588 05             DCR     B
D589 C285D5         JNZ     LSKP2
D58C C1             POP     B
D58D C9             RET

            ; OUTPUT TO TELEPHONE MODEM

            MOUT:   OUTPUT  M,Z
D58E+DB14   ??0004: IN      MSTAT   ;CHECK STATUAS
D590+E680           ANI     MOMSK   ;MASK FOR OUTPUT
D592+CA8ED5         JZ      ??0004  ;LOOP UNTIL READY
D595+79             MOV     A,C     ;GET BYTE
D596+D315           OUT     MDATA   ;OUTPUT IT
D598 C9             RET

D599 DB14   MIN:    IN      MSTAT   ;MODEM INPUT
D59B E640           ANI     MIMSK
D59D CA99D5         JZ      MIN
D5A0 DB15           IN      MDATA
D5A2 E67F           ANI     7FH     ;MASK PARITY
D5A4 C9             RET

            ; INPUT FROM LIST DEVICE

D5A5 DB12   LIN:    IN      LSTAT
D5A7 1F             RAR
D5A8 D2A5D5         JNC     LIN
D5AB DB13           IN      LDATA   ;KEEP 8 BITS
D5AD C9             RET

            ; FULL-DUPLEX MODEM

D5AE 79     MFULL:  MOV     A,C     ;GET BYTE

D5AF E67F           ANI     7FH     ;STRIP PARITY
D5B1 B7             ORA     A       ;NULL?
```

```
                     ; OUTPUT TO CONSOLE CRT

                     CONW:    OUTPUT  C,Z
D531+DB10            ??0001:  IN      CSTAT    ;CHECK STATUAS
D533+E602                     ANI     COMSK    ;MASK FOR OUTPUT
D535+CA31D5                   JZ      ??0001   ;LOOP UNTIL READY
D538+79                       MOV     A,C      ;GET BYTE
D539+D311                     OUT     CDATA    ;OUTPUT IT

                              IF      NNULS > 0
                              CPI     CR
                              RNZ

                     ; ADD DELAY AFTER CARRIAGE RETURN

                              PUSH    D        ;USE D,E
                              MVI     D,30 * NNULS
                     OUTCR:   MVI     E,250
                     OUTC2:   DCR     E
                              JNZ     OUTC2    ;INNER LOOP
                              DCR     D
                              JNZ     OUTCR    ;OUTER LOOP
                              POP     D
                              ENDIF

D53B C9                       RET

                     ; PUNCH OUTPUT

D53C 3A0300          PUNCH:   LDA     IOBYTE
D53F E630                     ANI     30H      ;BITS 4,5
D541 CAAED5                   JZ      MFULL    ;MODEM FULL DUPLEX
D544 B7                       ORA     A
D545 EA31D5                   JPE     CONW     ;30H IS CONSOLE
D548 E610                     ANI     10H      ;BIT 4
D54A C225D5                   JNZ     MHALF    ;10 H IS HALF DUPLEX
                              OUTPUT  L,Z      ;20H IS PUNCH (NO FF)
D54D+DB12            ??0002:  IN      LSTAT    ;CHECK STATUAS
D54F+E602                     ANI     LOMSK    ;MASK FOR OUTPUT
D551+CA4DD5                   JZ      ??0002   ;LOOP UNTIL READY
D554+79                       MOV     A,C      ;GET BYTE
D555+D313                     OUT     LDATA    ;OUTPUT IT
D557 C9                       RET

                     ; LIST OUTPUT

D558 3A0300          LOUT:    LDA     IOBYTE   ;CHECK IT
D55B E6C0                     ANI     0C0H     ;LEFT 2 BITS
D55D CA6BD5                   JZ      LIST     ;IF ZERO

                     ; LIST OUTPUT TO MODEM OR CONSOLE

D560 B7                       ORA     A

D561 EA25D5                   JPE     MHALF    ;MODEM, HALF DUPLEX
D564 17                       RAL              ;CHECK BIT 7
```

```
D5B2 C8                       RZ                ;DON'T SEND NULL
D5B3 FE0A                     CPI     LF        ;LINE FEED?
D5B5 C8                       RZ                ;DON'T SEND LINE FEED
D5B6 CD8ED5                   CALL    MOUT      ;SEND TO MODEM
D5B9 FE0D                     CPI     CR        ;CARRIAGE RETURN?
D5BB CAC5D5                   JZ      MODCR     ;WAIT FOR CR
D5BE CD99D5                   CALL    MIN       ;INPUT FROM MODEM
D5C1 4F                       MOV     C,A
D5C2 C328D5                   JMP     CONOUT    ;SEND TO CONSOLE

                     ; SEND A CARRIAGE RETURN AND WAIT FOR ONE

D5C5 CD99D5          MODCR:   CALL    MIN       ;INPUT FROM MODEM
D5C8 4F                       MOV     C,A
D5C9 CD28D5                   CALL    CONOUT    ;SEND TO CONSOLE
D5CC FE0D                     CPI     CR
D5CE C2C5D5                   JNZ     MODCR     ;LOOP UNTIL CR
D5D1 C9                       RET

D5D2 3739303430SIGNON:        DB      '790401'  ;REVISION DATE
                              IF      PERCOM
                              TITLE   'CPMIO 8080 Percom'
                              ELSE
                              TITLE   'CPMIO 8080 Mits'
                              ENDIF
                              IF      CLOCK
D5D8 2C20636C6B               DB      ', clk'
                              ENDIF

                     ; ******************************************
                              IF      INTRM    ;INTERRUPTS
D5DD 2C20696E74               DB      ', intr',LEV+'0'
                              ENDIF
                     ; ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

D5E4 00                       DB      0         ;MESSAGE END
D5E5 00              IFLAG:   DB      0         ;INITIALIZATION FLAG

D5E6                          END
```

```
00C5 ACONT     00C4 ADATA     00C7 BCONT     00C6 BDATA     EE06 BUFF
EE02 CCNT      0011 CDATA     0001 CIMSK     D41B CIN3      D442 CIN4
D440 CIN5      FFFF CLOCK     0002 COMSK     D412 CONIN     D528 CONOUT
D4F8 CONST     D531 CONW      EE00 CPNTR     000D CR        0010 CSTAT
0003 CTRC      0004 CTRD      0005 CTRE      000F CTRO      0011 CTRQ
0013 CTRS      0000 FALSE     000C FFEED     D5E5 IFLAG     D445 INIT
FFFF INTRM     D1E5 INTRP     0003 IOBYTE    EE03 KCNT      D4DF KEY2
D4CC KEY3      D4BA KEY4      D4DE KEY5      D4E2 KEY6      D4AA KEYBD
EE04 KPNTR     0013 LDATA     0005 LEV       000A LF        0001 LIMSK
D5A5 LIN       D51D LISST     D56B LIST      0002 LOMSK     D558 LOUT
D580 LSKIP     D585 LSKP2     0012 LSTAT     0015 MDATA     D49F MESG
D5AE MFULL     D525 MHALF     0040 MIMSK     D599 MIN       D5C5 MODCR
0080 MOMSK     D58E MOUT      0034 MSIZE     0014 MSTAT     0000 NNULS
D400 ORIGIN    FFFF PERCOM    D53C PUNCH     D51A QUIT      00F0 RESET
D4E8 RSETP     FFFF SHORT     D5D2 SIGNON    00FF SSW       D400 START
0095 STOPC     0011 STOPL     0007 TDATA     0001 TIMSK     0080 TOMSK
FFFF TRUE      0006 TSTAT     00FE VI
```

MICRO MARKET

MICRO-MARKET ADS SELL YOUR PRODUCTS

A new format has been established for the Micro-Market section. All ads are now 2 inches wide by 3 inches deep. Price is $200. Submit ads with check or money order to:

INTERFACE AGE Magazine
Micro-Market Ads
P.O. Box 1234
Cerritos, CA 90701
(213) 926-9544

We have discounts, manufacturer's warranties, FREE shipping and insurance *and* a TOLL FREE ORDER NUMBER available. *CALL US!*

Pan American Electronics
INCORPORATED
a **Radio Shack**® AUTHORIZED SALES CENTER
1117 Conway, Mission, Texas 78572
TOLL FREE ORDER NUMBER
800/531-7466
Texas & Principal No.
512/581-2765

master charge · VISA

CIRCLE INQUIRY NO. 90

ATTENTION: Apple Users!
WANT BETTER COLOR...MORE STABILITY?

Ever question the quality of your computer display. If so, you've probably been told "That's the best you can expect from an RF modulator...buy a color monitor".

WE CHALLENGE THAT STATEMENT!

DON'T BE SATISFIED WITH EXISTING QUALITY. See for yourself what our "new concepts" modulator can do for your picture...MICRO-VERTER Model MVX-500, $35 P.P. Phone orders welcomed.

HOTLINE DIAL: 402-987-3771

13-I BROADWAY **ATV Research** DAKOTA CITY, NE. 68731

DEALERSHIPS AVAILABLE
CATALOG UPON REQUEST

CIRCLE INQUIRY NO. 83

Connect your TRS-80, Apple or ANY other computer to the phone lines.

USR-330 Originate— Auto-Answer Modem

$339

- 0-300 Baud
- Stand Alone
- RS232
- 1 Year Warranty
- Crystal Controlled
- Bell 103/113
- State of the Art LSI circuitry
- 5 stage active filters

FCC certified for direct connection to phone lines via standard extension phone jack

Call or write for free literature

U.S. ROBOTICS, INC.
1035 W. LAKE ST.
CHICAGO, ILL. 60607
(312) 733-0497

CIRCLE INQUIRY NO. 91

SUMMER SPECIALS

VECTOR GRAPHIC SYSTEM B & PEACHTREE	$4295
APPLE II + 16K	$ 950
GRAPHICS TABLET	$ 659
LANGUAGE SYSTEM	$ 429
VISICALC	$ 125
BASF DISCS (10)	$ 28

PLEASE CALL FOR QUOTES ON APPLE PRODUCTS

DCC EQUIPMENT
11470 SOUTH STREET
CERRITOS, CA 90701
(213) 860-9441

CIRCLE INQUIRY NO. 86

BELOW DEALERS COST!

NorthStar
North Star Horizon
(FACTORY TESTED AND ASSEMBLED)
32 K DD – 2295.00
32 K QUAD – 2600.00
16K and a 32K RAM boards

Application Software

North Star BASIC:
- North Star Pascal $160.00
- NorthWord $335.00
- General Ledger $825.00
- Info Manager $395.00
- Mail Manager $245.00
- Accounts Receivable $475.00
- Accounts Payable $475.00

COD · CASH IN ADVANCE · VISA / MASTER CHARGE*
* ADD 2 PERCENT ON CHARGE ORDERS

CUSTOM BUSINESS COMPUTERS
103 ATLANTIC AVENUE, LYNBROOK, NEW YORK 11563 • (516) 887 1500 or 887 3340

CIRCLE INQUIRY NO. 85

NORTH STAR SOFTWARE
BUSINESS ACCOUNTING SYSTEMS

GENERAL LEDGER $300
250 Ledger Accounts
1500 Ledger Transactions

ACCOUNTS RECEIVABLE .$300
250 Accounts/2000 Transactions

ACCOUNTS PAYABLE $400
250 Vendors/1000 Transactions

PAYROLL $400
50 Accounts

INVENTORY $300
800 Accounts

500 CHECK REGISTER

COMPLETE BUSINESS SERVICES CORP.
90 WEST CENTER ST.
LOGAN, UTAH 84321
(801) 752-5713

ALL 5 PACKAGES $1500
1000 Checks
1000 Statements
500 Envelopes

CIRCLE INQUIRY NO. 84

Moonshadow Text Formatter for UCSD Pascal™ Systems

The UCSD Pascal™ system contains a screen-oriented text editor which is convenient, but which is not suited to word processing. It cannot underline, paginate automatically, or perform other essential text-processing functions. The **Moonshadow Text Formatter** (MTF) from Merrimack Systems solves this problem.

With the Moonshadow Text Formatter, documents produced with the screen editor are *post-processed* to provide these missing functions. It takes standard Pascal text files, operates on them, and sends fully formatted text output to the console display, a printer, or a disk file.

Moonshadow Text Formatter provides, in addition to a full range of formatting functions, *advanced features* including the combination of files into one document, variables in text (for form letters), and output character translation.

The Moonshadow Text Formatter is written in UCSD Pascal™, and is available for North Star Apple II, LSI-II, and 8080/Z80 systems with IBM Format 8" floppy disks.

All this for **$125.00** *from*
Merrimack Systems
POB 5218
Redwood City, CA 94063
(415) 365-6281

California residents should include 6% sales tax.

Also available: North Star Pascal personalization for SOL/SOLOS — just transfer three files and you're up with Pascal. $25.00

CIRCLE INQUIRY NO. 88

MEMOREX DISKETTES
Call toll FREE (800) 235-4137
PACIFIC EXCHANGES

3M DISKETTES
Call toll FREE (800) 235-4137
PACIFIC EXCHANGES

BASF DISKETTES
Call toll FREE (800) 235-4137
PACIFIC EXCHANGES

CIRCLE INQUIRY NO. 89

ADVERTISER INDEX

*Manufacturer requests factory-direct inquiry.

MICRODEX

MICRO INDEX CLASSIFIED

TRS-80 CLUBS Send name, address on club letterhead. We'll send a list of special club prices on drives, printers, etc., you'll find attractive. Aerocomp, Box 24829, Dallas, TX 75224.

NORTH STAR professional business software; Timeclok/billing; Office/Apt/Condo Management; Accounts Receivable. Each integrated with General Ledger. Single programs—$225 each; any two—$395. Write Ladco Development Co., P.O. Box 464, Olean, NY 14760, (716) 372-0168.

NORTH STAR compatible quad capacity disk controller. Tandon fast access mini disk drives. Disk cabinets and power supplies. CP/M compatible machine code generating BASIC compiler. Write or call for more information today. Midwest Digital, 863 Wood Ave., Wichita, KS 67212, (316) 721-1671.

CUSTOM CONTROL PANELS and switch plates for computers, industry, home or auto. TRS-80 Master Control Station, custom consoles and electronic controls. Free flyer. Custom Controls, 4 Fernandes Drive, So. Hadley, MA 01075.

QUALITY SOFTWARE. Database manager/report generator, advanced mailing list, inventory, word processor, GL, A/R, A/P and payroll systems. Available for TRS-80 Mod-I, II, CP/M and Heath. For catalog, send SASE (28¢) to Micro Architect Inc., 96 Dothan St., Arlington, MA 02174.

HEATH SOFTWARE. H-8/H-89 Disk Payroll System, Withholding, FICA, Straight State (PA), Local taxes. 50 employees. DLA Reprographics, 500 Golf Road, Lancaster, PA 17602, (717) 397-7406.

MICRODEX is a classified advertising section available to readers wishing to buy, sell or trade hardware and software. Price is $1 per word, with a 25 word minimum. The first two words bold. Additional bold type $5 per word.
Send copy with check or money order to INTERFACE AGE, Dept. M-I, 16704 Marquardt Ave., Cerritos, CA 90701.

INTERFACE AGE

Magazine Readership Survey

INTERFACE AGE would like to solicit the help of readers on surveying magazine readership. Please answer the following questions by circling the corresponding number of the adjacent reader service card.

Which, if any, of these magazines do you read regularly—that is at least three out of every four issues?

300 Popular Electronics
301 Datamation
302 Creative Computing
303 Computer Design
304 Byte
305 None of the above

Which of the following magazines do you subscribe to?

306 Popular Electronics
307 Creative Computing
308 Byte
309 None of the above

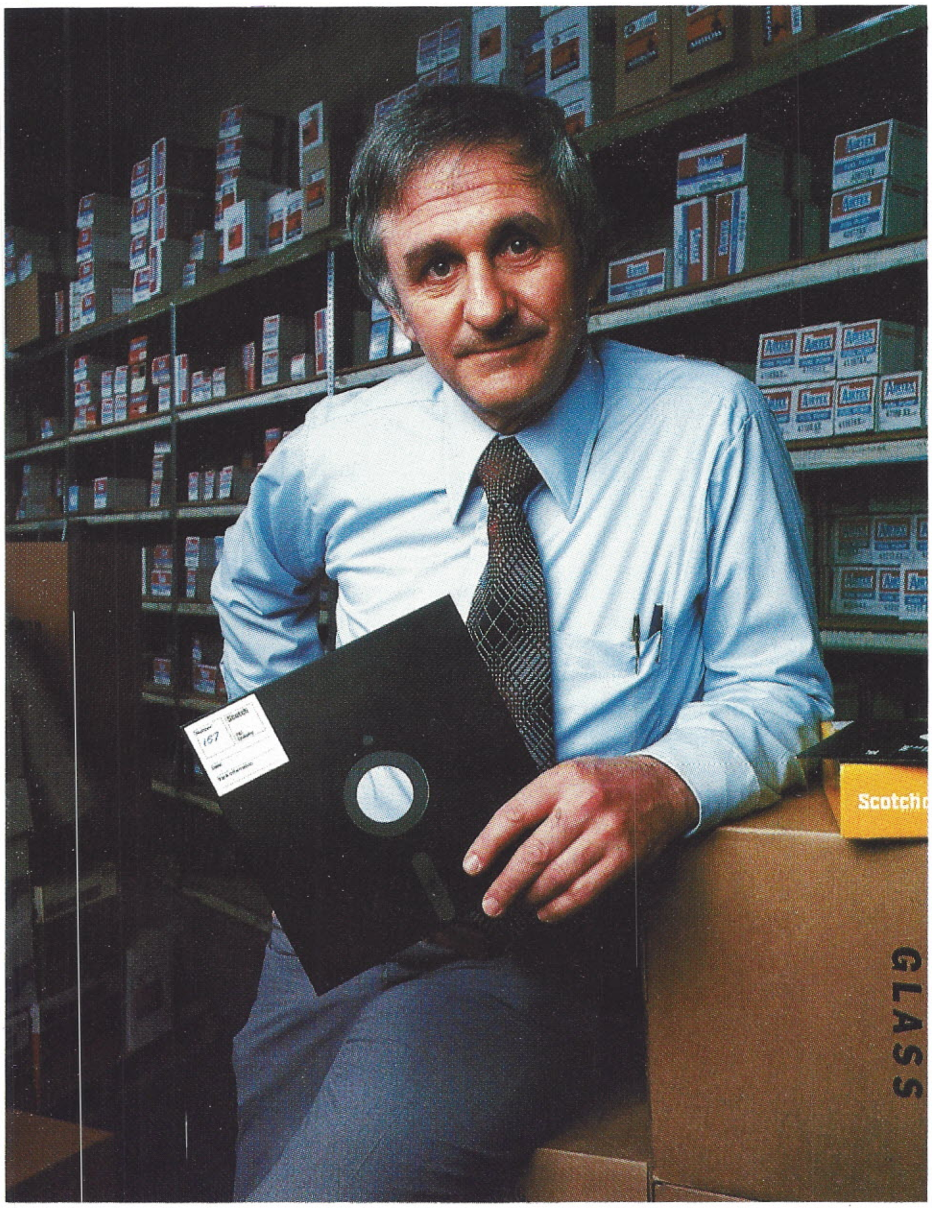
Which of the following magazines do you buy regularly at a computer store (at least three out of four issues)?

310 Popular Electronics
311 Creative Computing
312 Byte
313 None of the above

When purchasing computer-related products, which magazine do you turn to first as a buying reference?

314 Popular Electronics
315 Interface Age
316 Datamation
317 Creative Computing
318 Computer Design
319 Byte

"Our inventory is our existence. Think we'd trust it to anything less than Scotch® Brand Diskettes?"



Don Stone, President, Mass. Auto Supply Company, Inc., Boston, Mass.

Scotch Diskettes are the diskettes you can depend upon with the information *your* business depends upon.

Each one is tested and certified error-free before it leaves our factory. Because we know nothing less than perfection is acceptable for *your* vital business data.

Scotch Diskettes are available in regular or mini sizes, compatible with almost any system.

To find out where you can purchase Scotch Diskettes, call toll free: 800-328-1300. (In Minnesota, call collect: 612-736-9625.) Ask for the Data Recording Products Division. In Canada, write 3M Canada Inc., London, Ontario, N6A 4T1.

If it's worth remembering, it's worth Scotch Data Recording Products.

3M

CIRCLE INQUIRY NO. 33

Make your micro MIGHTY!

Get the most out of your microcomputer with Graham-Dorian Business software.

At any given time, your hardware is only as useful as the software you insert in it.

So it pays to rely on Graham-Dorian, the software that gets your micro performing to its fullest — almost like a mini.

Graham-Dorian, the industry leader, offers highly detailed and well-documented programs. All pretested on the job. Each so comprehensive that it takes little time to learn to run a program — even for someone who's never operated a computer before.

Programs are compatible with most major computers using CP/M disk operating systems, and come in standard 8″ or on various mini-floppy disks. Each package contains the software program in INT and BAS file form plus a user's manual and hard copy source listing. Graham-Dorian stands behind dealers with technical advice.

Yes, there's a world of difference in business software. Graham-Dorian has more per-package capabilities and more packages. (With new ones added every few months.)

The Graham-Dorian line now includes these packages:

- Medical
- Dental
- Surveying
- Inventory
- Payroll
- Apartment Management
- Construction Job Costing
- Accounts Receivable
- Accounts Payable
- General Ledger
- Cash Register

CBASIC-2

Ask your dealer for a demonstration soon.

G DS

Graham-Dorian Software Systems, Inc.

211 North Broadway / Wichita, KS 67202 / (316) 265-8633